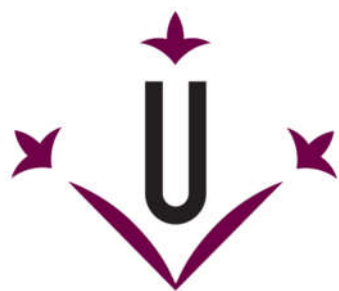


Sistema de adquisición de datos portátil para la generación de nubes de puntos 3D georreferenciadas a partir de un sensor LiDAR 2D, una unidad de medida inercial y un receptor GNSS-RTK

Máster en Ingeniería Industrial



Universitat de Lleida
Escola Politècnica Superior

Autor: Alejandro Solans Barón

Directores: Alexandre Escolà – Marcel Tresanchez

Septiembre 2017



Índice general

1. Introducción	7
1.1. Antecedentes.....	8
1.2. Objeto	11
2. Materiales y Métodos	12
2.1. Requisitos del diseño	12
2.1.1. Requisitos generales.....	12
2.1.2. Sensores a utilizar	12
2.1.3. Sistema de adquisición	14
2.2. Adquisición, preprocesamiento y almacenamiento de señales simultáneas 16	
2.2.1. Adquisición.....	16
2.2.2. Preprocesamiento.....	25
2.2.3. Almacenamiento	27
2.2.4. Señalización y botones de usuario	30
2.3. Posprocesamiento de datos: Generación de la nube de puntos	38
2.3.1. Preparación de los datos	38
2.3.2. Programa de apoyo al análisis	41
2.3.3. Generación de la nube de puntos	66
2.3.4. Validación de la nube de puntos	71
3. Resultados y discusión	73
3.1. Tiempos de adquisición y sincronización	73
3.2. Nubes de puntos.....	98
4. Conclusiones.....	105
5. Propuestas de mejora	106
6. Bibliografía	107
7. Anejos	108
7.1. Códigos de adquisición, preprocesamiento y guardado de datos	108
7.2. Códigos para el posprocesamiento de datos.....	111



Índice de figuras

Figura 1: Sensor LiDAR UTM-30LX-EW, del fabricante Hokuyo (Fuente: Hokuyo)	13
Figura 2: Unidad de medida inercial MTi, del fabricante Xsens (Fuente: Xsens)	13
Figura 3: Receptor Leica GPS 1200+, del fabricante Leica (Fuente: Leica)	13
Figura 4: Batería externa de 20000mAh del fabricante AUKEY	14
Figura 5: Señaladas las resistencias R68 y SB11 en la parte trasera de la placa STM32F4	15
Figura 6: Información seleccionada de la cadena enviada por el receptor GNSS-RTK	17
Figura 7: Evolución de la estructura de anclaje del sistema de adquisición desde su estado inicial (izquierda) a su estado final (derecha). En la parte superior se observa la antena del receptor GNSS-RTK, a media altura se aprecia el sensor LiDAR y en la parte inferior derecha, la unidad de medida inercial	19
Figura 8: Medidas del ángulo Roll tras la recepción de golpe incidente en el eje longitudinal del sensor registradas con una frecuencia de muestreo de 50Hz.	20
Figura 9: Guardado de un dato LiDAR, inicio de la muestra.	27
Figura 10: Guardado de un dato LiDAR, aparición de un multiretorno	28
Figura 11: Guardado de un dato LiDAR, final de muestra	28
Figura 12: Guardado de dato IMU	29
Figura 13: Guardado de dato GNSS-RTK	29
Figura 14: Diagrama vectorial de la declinación magnética	40
Figura 15: Figure estándar generado por Matlab	41
Figura 16: Figure Matlab adaptado	42
Figura 17: Menú Archivo	43
Figura 18: Mensaje de advertencia	43
Figura 19: Orden de apertura de nuevas figuras	44
Figura 20: Menú de preferencia de Matlab	45
Figura 21: Menú Gráficos	45
Figura 22: Menú Gráficos-NUBE	46
Figura 23: Nube de puntos, visión 3D	46
Figura 24: Nube de puntos, plano de corte XY	47
Figura 25: Nube de puntos, plano de corte XZ	47
Figura 26: Nube de puntos, plano de corte YZ	48
Figura 27: Menú Gráficos-LiDAR	48
Figura 28: Selección del scan a analizar	49
Figura 29: Mensaje de advertencia, scan no alcanzable, no se han realizado tantos	49
Figura 30: Graficado de todos los datos LiDAR recogidos	50
Figura 31: Graficado del primer rebote LiDAR	50
Figura 32: Graficado del segundo rebote LiDAR	51
Figura 33: Menú Gráficos-GNSS	52
Figura 34: Graficado de los datos del receptor GNSS-RTK, corte del plano XY	52
Figura 35: Graficado de la coordenada X del receptor GNSS-RTK	53
Figura 36: Graficado de la coordenada Y del receptor GNSS-RTK	53
Figura 37: Graficado de la coordenada Z del receptor GNSS-RTK	54
Figura 38: Menú Gráficos-IMU	54
Figura 39: Graficado de los tres ángulos de la unidad de medida inercial	55



Figura 40: Graficado del ángulo roll de la unidad de medida inercial	56
Figura 41: Graficado del ángulo pitch de la unidad de medida inercial.....	56
Figura 42: Graficado del ángulo yaw de la unidad de medida inercial	57
Figura 43: Menú Herramientas	57
Figura 44: Menú de opciones de la herramienta desplazar.....	58
Figura 45: Visualización de datos con el cursor de datos	59
Figura 46: Visualización de datos múltiples con el cursor de datos.....	60
Figura 47: Menú Análisis temporal	60
Figura 48: Menú Análisis temporal-LiDAR.....	61
Figura 49: Graficado del tiempo muerto del sensor LiDAR.....	61
Figura 50: Graficado del tiempo de procesado y guardado del sensor LiDAR.....	62
Figura 51: Tiempo entre recepciones del sensor LiDAR.....	63
Figura 52: Menú Análisis temporal-GNSS	63
Figura 53: Tiempo entre recepciones del receptor GNSS-RTK.....	64
Figura 54: Menú Análisis temporal-IMU	64
Figura 55: Tiempo entre recepciones de la unidad de medida inercial	65
Figura 56: Explicación gráfica de los ángulos roll, pitch y yaw.....	67
Figura 57: MTi en el sistema de coordenadas de la Tierra(Fuente:Xsens)	68
Figura 58: Matriz de rotación para ángulos de Euler	68
Figura 59: Diagrama de flujo de la función EulerAngles	69
Figura 60: Figuras geométricas desde diferentes ángulos preparadas para la primera prueba de validación	71
Figura 61: Opel Astra 5p (2004)	72
Figura 62: Tiempo entre recepciones del sensor LiDAR [ND - sin skips].....	74
Figura 63: Tiempo de procesado y guardado de sensor LiDAR [ND - sin skips]	74
Figura 64: Tiempo entre recepciones del sensor LiDAR [ND – 3 skips].....	75
Figura 65: Tiempo de procesamiento y guardado de sensor LiDAR [ND – 3 skips]	76
Figura 66: Tiempo entre recepciones del sensor LiDAR [ND – 9 skips].....	77
Figura 67: Tiempo de procesamiento y guardado de sensor LiDAR [ND - 9 skips]	77
Figura 68: Tiempo entre recepciones del sensor LiDAR [ND – 1 skip (sin guardado)]	78
Figura 69: Tiempo de guardado de timer en SD [ND – 1 skip]	79
Figura 70: Tiempo entre recepciones del sensor LiDAR [ND – 3 skips (sin guardado)]	79
Figura 71: Tiempo de guardado de timer en SD [ND – 3 skips]	80
Figura 72: Tiempo entre recepciones de sensor LiDAR [HD]	81
Figura 73: Ampliación del eje Y de la gráfica de la figura 72.....	82
Figura 74: Tiempo de procesado y guardado de sensor LiDAR [HD]	82
Figura 75: Ampliación del eje Y de la gráfica de la figura 74.....	83
Figura 76: Tiempo muerte del sensor LiDAR [HD].....	83
Figura 77: Tiempo entre recepciones de la unidad de medida inercial	84
Figura 78: Ampliación del eje Y de la gráfica de la figura 77.....	85
Figura 79: Tiempo entre recepciones del receptor GNSS-RTK.....	85
Figura 80: Ampliación del eje Y de la gráfica de la figura 79.....	86
Figura 81: Tiempo entre recepciones del sensor LiDAR [Adq. simult.- Objetivo 1]	87
Figura 82: Tiempo de procesado y guardado de datos del sensor LiDAR [Adq. simult.- Objetivo 1]	88



Figura 83: Tiempo muerto del sensor LiDAR [Adq. simult.- Objetivo 1]	89
Figura 84: Tiempo entre recepciones de la unidad de medida inercial [Adq. simult.- Objetivo 1]	90
Figura 85: Ampliación del eje Y de la gráfica de la figura 80.....	90
Figura 86: Tiempo entre recepciones del receptor GNSS-RTK [Adq. simult.- Objetivo 1]	91
Figura 87: Ampliación del eje Y de la gráfica de la figura 86.....	92
Figura 88: Tiempo entre recepciones del sensor LiDAR [Adq. simult.- Objetivo 2]	93
Figura 89: Ampliación del eje Y de la gráfica de la figura 88.....	94
Figura 90: Tiempo de procesado y guardado de datos del sensor LiDAR [Adq. simult – Objetivo 2]	95
Figura 91: Tiempo muerto del sensor LiDAR [Adq. simult.- Objetivo 2]	95
Figura 92: Tiempo entre recepciones de la unidad de medida inercial [Adq. simult.- Objetivo 2]	96
Figura 93: Tiempo entre recepciones del receptor GNSS-RTK [Adq. simult.- Objetivo 2]	97
Figura 94: Nube de puntos, rotación completa al vehículo [Objetivo 1]	98
Figura 95: Nube de punto, giro yaw de 360° desde punto fijo [Objetivo 1].....	99
Figura 96: Nube de puntos, giro pitch desde punto fijo [Objetivo 1]	99
Figura 97: Nube de puntos, lateral izquierdo vehículo [Objetivo 1]	100
Figura 98: Nube de puntos, lateral derecho vehículo [Objetivo 1].....	100
Figura 99: Nube de puntos, lateral izquierdo delantero vehículo [Objetivo 2]	101
Figura 100: Nube de puntos, lateral izquierdo trasero vehículo [Objetivo 2]	102
Figura 101: Nube de puntos, figuras geométricas [Objetivo 2]	103



Agradecimientos

Detrás de este trabajo hay personas que han guiado mis pasos ya sea con conocimiento o con su incondicional apoyo. Sin ellas la realización de este proyecto no hubiese sido posible y por ello me gustaría dedicarles unas líneas con las que hacerles llegar mi más sincero agradecimiento.

En primer lugar, a todos los miembros del Grup de Recerca en AgròTICa i Agricultura de Precisió (GRAP) de la Universitat de Lleida, por haberme hecho sentir como uno más dentro del grupo, y haber depositado su confianza en mí poniendo el proyecto en mis manos.

Especial agradecimiento dentro de este grupo a Àlex Escolà, por confiar en mí y guiar mis investigaciones durante ya más de un par de años, ya que no es fácil confiar en que algo vaya a acabar haciendo su función si semana sí semana también falla una u otra cosa y todo parecen “excusas”. En parte por no fallarme a mí y en parte por no fallar a su confianza he luchado hasta conseguir los resultados.

A todos los profesores y compañeros que en algún momento me han regalado parte de sus conocimientos poniendo así un escalón más para llegar al objetivo, Marcel Tresanchez, Jordi Llorens, Francisco Clariá, Jordi Palacín, Josep Ribó, Isaac García, Sara Vergés; y en general a todos aquellos que de alguna forma se han interesado por mi proyecto.

Y por último, un profundo agradecimiento a mi familia, pareja y amigos, por estar siempre allí, sin su apoyo y comprensión en los momentos más duros no hubiese llegado hasta el final en ésto.

En especial, un agradecimiento a mi abuelo Ramón Barón, que siempre mostró admiración y respeto por mi trabajo. Te lo dedico estés donde estés.



1. Introducció

El presente proyecto surge de la idea de poder mejorar un sistema para obtener los parámetros geométricos y estructurales (fenotipo) de plantas y árboles en campo, a demanda del Grup de Recerca en AgròTICa i Agricultura de Precisió (GRAP) de la Universitat de Lleida. El GRAP empezó sus trabajos en caracterización electrónica de la vegetación mediante sensores LiDAR (light detection and ranging) en el año 2002 y actualmente dispone de un escáner láser terrestre móvil de diseño propio basado en un sensor LiDAR 2D, un sistema de posicionamiento global basado en satélites y Real-Time Kinematics (RTK) y un sistema de adquisición de datos basado en un programa desarrollado en LabVIEW (National Instruments) ejecutado en ordenador portátil de campo.

Con el sistema actual se consiguen nubes de puntos 3D, pero se tienen algunos inconvenientes, los cuales se quieren solventar en este nuevo proyecto, para lo cual se decide añadir una Unidad de Medida Inercial (en adelante IMU), que nos servirá para conocer en todo momento las inclinaciones en los 3 ejes del sensor. Además, se pretende cambiar el sistema de adquisición de datos, transportando este trabajo a un microcontrolador de gama alta para desarrollar un escáner láser portátil. También se desarrolla un programa de posprocesamiento para realizar la construcción y posterior visualización de la nube de puntos tridimensional.



1.1. Antecedentes

El Grupo de Investigación en AgróTICa y Agricultura de Precisión (GRAP) nació en 2002 aglutinando investigadores de dos organismos: la Universitat de Lleida (UdL) y el Centro de Mecanización Agraria del Departamento de Agricultura, Ganadería, Pesca, Alimentación y Medio Natural de la Generalitat de Catalunya. Sin embargo, los investigadores integrantes llevan más de 15 años trabajando conjuntamente y constituyen uno de los grupos de referencia en Tecnología de Aplicaciones Fitosanitarias a nivel de todo el Estado español, con varias patentes y modelos de utilidad en este ámbito. El grupo también es pionero en el ámbito de los equipos robotizados e inteligentes para la ganadería de precisión, con una de las primeras patentes en este ámbito y equipos instalados en Canadá.

AgróTICa es un término que hace referencia a la aplicación de las Tecnologías de la Información y la Comunicación (TIC) en la agricultura, en sentido amplio. Las aplicaciones de las TIC pueden ser muchas y diversas. Así, las TIC pueden utilizarse para adquirir datos de sistemas agrícolas o ganaderos, convertir los datos en información útil para los agricultores y/o ganaderos y comunicar y/o compartir esta información mediante sistemas de comunicación.

La **Agricultura de Precisión** consiste en realizar las operaciones agrícolas adecuadas, de la manera adecuada, en el momento adecuado, en el lugar adecuado y de la manera adecuada (adaptado de las definiciones de Pierre Robert y Raj Khosla). Para poder llevar a cabo esta agricultura es necesario conocer el comportamiento de los campos y sus peculiaridades para obtener de él el máximo provecho pero de forma eficiente y sostenible. Actualmente, los avances tecnológicos y las TIC permiten recopilar muchos datos sobre el cultivo y su medio con una altísima resolución espacial (muchos datos por unidad de superficie) y con un coste razonable. Visualizar estos datos, procesarlos y convertirlos en información útil, permite que los agricultores dispongan de un apoyo objetivo y fiable para poder tomar decisiones de manera coherente.



La práctica de la Agricultura de Precisión se puede llevar a cabo según dos grandes metodologías:

- Agricultura de Precisión basada en mapas digitales de información:

Antes de cualquier operación agrícola es necesario adquirir datos de la parcela, analizar su variabilidad espacial y mapearlos (crear un mapa de la parcela con la distribución espacial de la variable medida). Después de analizar los mapas obtenidos debe tomarse una decisión de manejo. El resultado de la toma de decisión será un nuevo mapa, denominado mapa de actuación o de prescripción, donde se muestra qué debe hacerse en cada punto de la parcela (intensidad de la operación o dosis de producto a aplicar). Para practicar este tipo de agricultura es indispensable disponer de un sistema de posicionamiento y navegación (vulgarmente llamado “un GPS”) más o menos preciso tanto para la adquisición de datos como para la actuación.

- Agricultura de Precisión basada en sensores y en tiempo real:

Este tipo de agricultura estrictamente no requiere sistemas de posicionamiento y navegación puesto que la adquisición de datos, la decisión y la actuación se llevan a cabo en tiempo real mientras el tractor y el equipo se van desplazando por la parcela. Dado que la actuación va a seguir siendo variable, el equipo también debe ir equipado con tecnología de actuación variable. La diferencia radica en que la actuación no se basa en un mapa de prescripción sino en uno o varios sensores que van adquiriendo datos “sobre la marcha”.

Basándose en el primer tipo de Agricultura de Precisión, el GRAP, con la idea de obtener modelos 3D y mapas de la vegetación agrícola, se embarcó en un proyecto que consistía en desarrollar un escáner láser terrestre móvil (MTLS) basado en un sensor LiDAR 2D y un receptor RTK, adquiriendo estos datos a través de un ordenador. A este proyecto se le añadió una plataforma móvil basada en orugas y un sistema para mantener su horizontalidad basado en un sistema de realimentación de primer orden.

Este sistema ya se ha puesto en marcha y con él se han conseguido construir diferentes modelos y mapas, un ejemplo de los resultados conseguidos se puede apreciar en Escolà et al., (2017), pero el objetivo actual es construir un sistema portátil a la vez que se mejora la



precisión y funcionalidad de éste, así como la creación de un código capaz de realizar el procesado de los datos adquiridos y convertirlos en una nube de puntos tridimensional de coordenadas cartesianas.

En 2015 se presentó como Trabajo de Final de Grado *“Sistema de adquisición de datos portátil para la generación de nubes de puntos 3D georreferenciadas a partir de un sensor LiDAR 2D (Parte I, Comunicación con los sensores) [2015, Solans, A.]”*, donde se realizó un estudio de viabilidad para la realización de esta transformación y, tras ello, se realizó la comunicación de los tres sensores individualmente.



1.2. Objeto

El Grup de Recerca en AgròTICa i Agricultura de Precisió ha construïdo un sistema que consta de un escàner làser terrestre mòbil (MTLS) basat en un sensor LiDAR 2D i un sistema satel·lital de navegació global Real-Time Kinematics (GNSS-RTK), junt amb un sistema de adquisició de dades que consisteix en un ordinador portàtil amb un software propi elaborat en LabVIEW, per a l'obtenció del fenotip de la vegetació del camp estudiat. Tot això està implementat sobre una plataforma d'oruga mòbil, la qual els permet desplaçar-se a través de les hileres d'arbres a estudiar. Per mantenir el sensor horitzontal, han implementat un sistema dinàmic d'estabilització tipus *gimbal* per a la correcció de posició.

El objecte d'aquest projecte és millorar els punts febles d'aquest sistema, com són la baixa manejabilitat del sistema, problemes del sistema d'estabilització degut a les vibracions del equip, dificultat en el maneig del software d'adquisició; i així aconseguir un sistema portàtil més compacte, manejable, lleuger, precís i econòmic, així com donar un pas més en la funcionalitat. També es desenvoluparà un programa per realitzar la reconstrucció de la núvol de punts georreferenciada a partir dels dades obtinguts.

Per a això, i seguint el treball realitzat en el Treball de Final de Grau "*Sistema de adquisició de dades portàtil per a la generació de nubes de punts 3D georreferenciades a partir d'un sensor LiDAR 2D (Parte I, Comunicació amb els sensors)*" [2015, Solans A.]":

- Se sincronitzarà la adquisició de dades dels tres sensors conjuntament, així com el emmagatzematge dels tres sensors en una memòria SD de forma ordenada.
- Se gestionarà el preprocesament de dades, així com el emmagatzematge en la memòria d'aquests per a que tinguin el mínim desfase temporal possible.
- Se treballarà en una interfície gràfica en Matlab (una per a cada sensor individual i altra per a les tres sincronitzades), que permeti observar amb major facilitat els dades obtinguts, així com els temps necessaris per a cada part del procés.
- Se desenvoluparà, en la interfície de Matlab preparada per gestionar els dades dels tres sensors sincronitzats, un algoritme que reconstrueixi la núvol de punts i guardi els dades de forma correcta per a que puguin ser visualitzats posteriorment amb el programa Cloud Compare.



2. Materiales y Métodos

2.1. Requisitos del diseño

Los requisitos del diseño vendrán dados por diferentes condicionantes. El primero de ellos es el entorno de utilización del sensor, en este caso el agrícola; por otro lado se tendrá, como en todo elemento de medida, requisitos relacionados con la velocidad y la precisión en la adquisición de datos; y por último, las limitaciones marcadas por los elementos a utilizar, ya que son elementos ya adquiridos y hay que adaptarse a ellos al 100%.

2.1.1. Requisitos generales

Por la parte mecánica, se requiere un diseño cómodo, ligero y manejable; a la vez, tiene que ser resistente y seguro para nuestros sensores, los cuales tienen que estar bien anclados para evitar vibraciones.

Por la parte electrónica, se requiere que el sistema sea capaz de captar, procesar y guardar los datos de los 3 sensores a una frecuencia máxima de 40 Hz, ya que es el límite de transmisión de datos impuesto por el sensor LiDAR y éste será el sensor de referencia del que no se quiera perder información.

Por otra parte, se requiere que el sistema sea fácil de utilizar y entendible a la hora de trabajar con él, es decir, conocer en todo momento su estado; y robusto, es decir, que no pierda datos ni se bloquee.

2.1.2. Sensores a utilizar

Para la realización de este proyecto se parte de 3 sensores ya adquiridos:

- LiDAR modelo UTM-30LX-EW, del fabricante Hokuyo, el cual se muestra en la figura 1, encargado de proporcionar datos de distancia.



Figura 1: Sensor LiDAR UTM-30LX-EW, del fabricante Hokuyo (Fuente: Hokuyo)

- Unidad de medida inercial modelo MTi, del fabricante Xsens, la cual se muestra en la figura 2, encargada de proporcionar los datos de posicionamiento espacial.



Figura 2: Unidad de medida inercial MTi, del fabricante Xsens (Fuente: Xsens)

- Receptor GNSS-RTK modelo Leica GPS 1200+, del fabricante Leica, el cual se muestra en la figura 3, encargado de proporcionar la geolocalización.



Figura 3: Receptor Leica GPS 1200+, del fabricante Leica (Fuente: Leica)

2.1.3. Sistema de adquisición

El sistema de adquisición está formado por:

- La placa electrónica STM32F4-Discovery del fabricante STMicroelectronics con el microcontrolador ARM Cortex-M4 STM32F407VGT6 de 32 bits.
- La placa electrónica de expansión correspondiente, utilizada para la comunicación Ethernet (para la comunicación con el sensor LiDAR), la comunicación RS-232 (para la comunicación con el sensor inercial) y la comunicación con la tarjeta SD.
- Un convertidor MAX232 para la comunicación RS-232 con el receptor GNSS-RTK.

Se necesita alimentar el sistema de adquisición de forma externa con una batería externa de 20000mAh como la que se muestra en la figura 4.



Figura 4: Batería externa de 20000mAh del fabricante AUKEY

Para ello, en la placa principal, STM32F4-Discovery, es necesario desoldar los siguientes componentes con el fin de desconectar permanentemente varias líneas del programador que van hacia el microcontrolador:

- R68: Para desconectar la línea MCO.
- SB11: Para desconectar el NRST del programador.

Estos dos componentes se ven señalados en la figura 5.



2.2. Adquisición, preprocesamiento y almacenamiento de señales simultáneas

Para conseguir el completo funcionamiento de este proyecto se necesita la adquisición de datos de los tres sensores, uno que mide la distancia entre el objeto y el sensor, el segundo que obtiene la orientación espacial de nuestro sensor, y por último un sensor que determine la localización exacta en el espacio desde donde se está realizando la medida. Por otro lado, se necesitará un dispositivo para realizar el guardado de todos estos datos para después poder utilizarlos.

Todo esto se debe realizar en tiempo real, obteniendo y guardando los datos de los tres sensores sin perder la capacidad de envío de datos de éstos e intentando obtenerlos en el tiempo que se reciben y no más tarde, ya que ambas cosas afectarán a la precisión del sistema.

Una vez adquiridos los datos de estos tres sensores y guardados correctamente, se podrán trabajar en un entorno como Matlab y se podrá dibujar un mapa en tres dimensiones del entorno escaneado.

Se tratarán por separado las acciones de adquisición, preprocesamiento y almacenamiento, aunque se debe tener en cuenta que los cambios en cualquiera de ellas afecta a las otras dos y que un cambio en una de ellas supone adaptar las otras dos a dicho cambio. Es decir, una vez analizadas las posibles soluciones para los tres procesos por separado, no es válida la suposición de que juntando las “tres mejores soluciones” se vaya a obtener la solución general óptima, sino que hay que unir las y configurarlo de tal forma que se encuentre la solución deseada.

2.2.1. Adquisición

En este apartado se tratará lo que respecta a la comunicación con los tres sensores y la recepción de la información que éstos envían, las diferentes formas estudiadas para que la simultaneidad de los sensores sea la deseada, la reducción de tiempo y errores en el procesamiento y las instrucciones más eficientes a la hora de recibir los datos, con sus respectivas consecuencias en los procesos posteriores.



Se comenzará con las mejoras del código que afectan individualmente a cada sensor, ya que la comunicación individual ya funcionaba correctamente, pero se ha detectado la necesidad de gestionar los tiempos de una forma más eficiente para que la comunicación sea más rápida y así evitar errores de bloqueo y también mejorar la precisión de las medidas adquiridas.

Receptor GNSS-RTK

En lo que respecta al receptor Leica GNSS-RTK, las medidas más importantes realizadas son:

- Reducir la cantidad de caracteres que guarda, es decir, almacenar tan solo los datos necesarios (coordenadas). En la figura 6 se muestra la selección de datos realizada.

\$--LLQ,hhmmss.ss,mmddyy,eeeeee.eee,M,nnnnnn.nnn,M,x,xx,x.x,x.x,M*hh<CR><LF>	
Campo	Descripción
\$--LLQ	Encabezado, incluyendo ID del transmisor
hhmmss.ss	Tiempo UTC de la posición
mmddyy	Fecha UTC
eeeeee.eee	Este de cuadrícula en metros
M	Unidades para el Este de cuadrícula, como texto fijo M
nnnnnn.nnn	Norte de cuadrícula en metros
M	Unidades para el Norte de cuadrícula, como texto fijo M
x	Calidad de posición 0 = Fija no disponible o no válida 1 = Sin posición en Tiempo Real, Fija de navegación 2 = Posición en Tiempo Real, ambigüedades sin fijar 3 = Posición en Tiempo Real, ambigüedades fijas
XX	Número de satélites empleados en el cálculo
x.x	Calidad de coordenadas en metros
x.x	Altitud de la posición del marcador sobre/debajo del nivel medio del mar en metros. En caso de no disponer de valores de altura ortométrica, se exportará la altura elipsoidal local.
M	Unidades de altitud, como texto fijo M
*hh	Suma de control
<CR>	Carriage Return
<LF>	Line Feed

Figura 6: Información seleccionada de la cadena enviada por el receptor GNSS-RTK

- Aumentar la velocidad de transmisión de datos, de 19200 a 115200 baudios, para así reducir el tiempo de transmisión y, por lo tanto, el tiempo que el microcontrolador está pendiente de este sensor.
- Se eliminan datos viejos cuando se comienza a recibir un nuevo dato, es decir, se limpia el buffer de entrada si éste no ha sido consultado y recibe una nueva posición del dispositivo.



- Limpieza y optimización del código eliminando líneas innecesarias y realizando algunas acciones de una forma menos costosa a nivel de número de procesos.
- Modificación del código de color de los led de la placa electrónica STM32F4-Discovery para saber en todo momento el estado del programa, eliminando de esta forma uno de los puertos USART (apartado 2.2.4 *Señalización y botones de usuario*).

Sensor de medida inercial

En lo relativo al sensor de medida inercial, se han realizado cambios tanto en lo respectivo al software, como a su disposición, como al hardware, pero sobretodo se ha comprobado su fiabilidad bajo diferentes situaciones y entornos:

- Se aumenta la velocidad de transmisión de 115200 a 916200 baudios.
- Se corrobora que la velocidad máxima de transmisión se obtiene cuando el sensor envía solo los ángulos en los tres ejes (roll, pitch y yaw).
- Se aleja de cualquier tipo de metal ferromagnético, ya que se ha comprobado que puede afectar de forma significativa a las lecturas de rumbo (yaw). Con el sensor de campo magnético de un Smartphone (One Plus One) se ha comprobado que el campo magnético local es débil, y que cualquier material ferromagnético cercano es capaz de desvirtuarlo en un porcentaje muy significativo.
- Como se puede observar en la figura 7, el sensor inercial (sensor naranja que se ve en la foto) pasa de estar sobre una estructura con tornillos y varas roscadas de acero (en el interior de las viguetas de aluminio), a estar en una estructura formada solo por aluminio. También se puede observar que los tornillos de acero que utilizan de base la antena receptora GNSS-RTK y el sensor LiDAR para su anclaje se han alejado lo más posible del sensor inercial.



Figura 7: Evolución de la estructura de anclaje del sistema de adquisición desde su estado inicial (izquierda) a su estado final (derecha). En la parte superior se observa la antena del receptor GNSS-RTK, a media altura se aprecia el sensor LiDAR y en la parte inferior derecha, la unidad de medida inercial.

- La solución no puede ser aislarlo completamente ya que entonces también se aislaría del campo magnético local.
- Se comprueba que las lecturas tras un golpe son erróneas hasta que los acelerómetros se vuelven a estabilizar, con lo que el sistema no debería recibir golpes mientras se realiza la lectura.
- En la figura 8 se puede observar lo nombrado anteriormente tras la recepción de un leve golpe. La línea azul es el momento del impacto, la línea verde el momento de estabilización, y el área sombreada en naranja es el tiempo que el sensor estaría enviando datos erróneos.

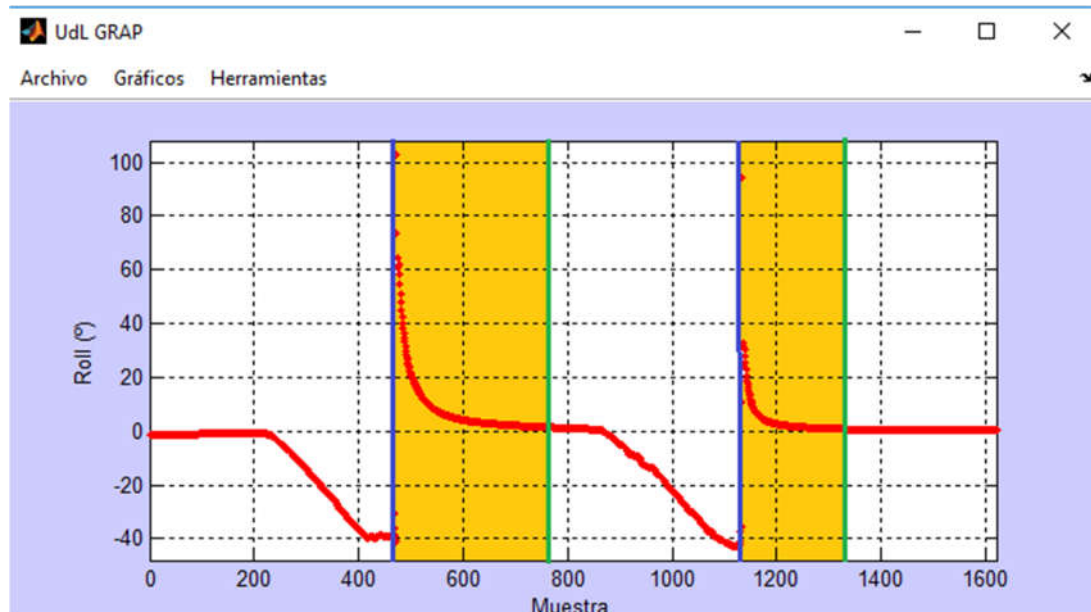


Figura 8: Medidas del ángulo Roll tras la recepción de golpe incidente en el eje longitudinal del sensor registradas con una frecuencia de muestreo de 50Hz.

- Se eliminan datos viejos cuando se comienza a recibir un nuevo dato, es decir, se limpia el buffer de entrada si éste no ha sido consultado y recibe una nueva orientación del dispositivo.
- Se analiza la concatenación de datos como método para guardar todos los datos en una sola orden (en el apartado 2.2.3. *Almacenamiento* se analiza el posible error de almacenamiento).
- Se limpia y optimiza el código eliminando líneas innecesarias y realizando algunas acciones de una forma menos costosa a nivel de número de procesos.
- Se modifica el código de color de los led de la placa para saber en todo momento el estado del programa, eliminando de esta forma uno de los puertos USART (apartado 2.2.4 *Señalización y botones de usuario*).



Sensor LiDAR

En lo referente al sensor LiDAR, se ha realizado un arduo trabajo en la mejora del software, probando todas las posibilidades encontradas para evitar el bloqueo del software así como intentando mejorar el tiempo de adquisición.

- Pruebas con el comando de recibo de datos tras petición (comando HD)

- Cambio de timers

Se realizan diferentes pruebas cambiando los tiempos de espera para pedir un nuevo dato, buscando preferentemente evitar el bloqueo del código cuando éste lleva un tiempo recogiendo datos. Se concluye que en caso de recoger datos tan solo del LiDAR, habría que esperar unos 200ms a coger una nueva medida para que el bloqueo sea improbable, aunque los datos nos revelan que esperando 100ms no debería haber problema e incluso podríamos movernos en esperas de 50-60ms.

De todas formas, por algún motivo que por el momento se desconoce, se cree que tiene que ver con la gestión de la memoria, si se reduce a estos tiempos el programa acaba bloqueándose cuando lleva alrededor de un minuto adquiriendo datos, en ocasiones más en ocasiones menos; no tiene un comportamiento estable, con lo que, para el objeto de este proyecto, que es el de la reconstrucción de la nube de puntos, se ha decidido trabajar con los 200ms de espera entre peticiones.

Ésto puede cambiar al recoger los datos de todos los sensores simultáneamente, como se analizará al final de este apartado.

- Eliminación carácter “;”

Se encuentra un error en el envío de la orden HD, en el cual al final de la petición se asumía que había que poner un “;” ya que cuando se realizó el anterior proyecto se confundió lo que decía en la guía. Ese punto y coma es para añadir algún tipo de comentario. Se procede a eliminarlo para ganar tiempo.



- Conclusiones en el uso del comando HD
 - Se ha conseguido que el código no se bloquee por mucho tiempo que esté corriendo.
 - El tiempo de recepción de datos es estable, sin grandes saltos como ocurre cuando se quiere ir demasiado rápido.
 - Se debe conseguir acelerar el proceso para obtener más datos por segundo y, en consecuencia, una nube de puntos más definida.
- Pruebas con uso del comando de recibo continuo de datos (comando ND)
 - Envío salteado

El microcontrolador no es capaz de recibir y procesar los datos tan rápido, es por eso que se explora la posibilidad de enviar los datos del LiDAR en vez de cada vez que escanea, saltando alguno de sus escaneos. Por el momento la opción no funciona 100%, ya que sí se consigue que el programa deje de bloquearse, pero se reciben los datos cada casi medio minuto.
 - Eliminación carácter “;”

Se encuentra un error en el envío de la orden HD, en el cual al final de la petición se asumía que había que poner un “;” ya que cuando se realizó el anterior proyecto se confundió lo que decía en la guía. Ese punto y coma es para añadir algún tipo de comentario. Se procede a eliminarlo para ganar tiempo.
 - Conclusiones con el uso del comando ND
 - Se consigue evitar el bloqueo aplicando saltos entre escaneos del LiDAR.
 - Eliminando el preprocesamiento los escaneos llegan al tiempo que tocan pero incluyendo el preprocesamiento los datos llegan cada casi medio minuto, con lo cual por el momento el comando ND no es una opción viable. Se analizarán las opciones con y sin preprocesamiento en el apartado con ese nombre que viene a más adelante.



- Se analiza la concatenación de datos como método para guardar todos los datos en una sola orden (en el apartado 2.2.3. *Almacenamiento* se analiza el posible error de almacenamiento).
- Se limpia y optimiza el código eliminando líneas innecesarias y realizando algunas acciones de una forma menos costosa a nivel de número de procesos.
- Se modifica el código de color de los led de la placa para saber en todo momento el estado del programa, eliminando de esta forma uno de los puertos USART (apartado 2.2.4 *Señalización y botones de usuario*).

Todos estos cambios llevan a una conclusión, que es que la mejor solución es utilizar el comando HD para la recepción de datos, cuya configuración final dependerá también de los análisis que se realizarán en los apartados posteriores de este informe.

Los tres sensores en adquisición simultánea

Una vez analizadas las mejoras individuales, se analiza el comportamiento simultáneo de recogida de datos de los 3 sensores.

En un principio, lo que se propuso fue una recogida “aleatoria”, es decir, cuando un sensor había enviado un dato, el microcontrolador lo recogía y volvía a esperar otro dato de cualquiera de los sensores. Por el momento se ha decidido descartar este método ya que provocaba problemas en la recepción del que se considera el sensor más significativo, como es el LiDAR.

Lo que se decidió fue estar a la espera del dato LiDAR y, una vez recibido, consultar la posición y orientación de los otros dos sensores. Es posible que se pierda un mínimo de precisión, la cual se intentará resolver con el posprocesamiento con Matlab, pero este método, con las limitaciones temporales que se experimentan, es el más efectivo para evitar errores de bloqueo del dispositivo, ya que la comunicación con el LiDAR es muy pesada y tener activo el procesamiento y guardado de los otros dos sensores se cree que es la causa del bloqueo por



falta de memoria en el dispositivo (es tan solo una suposición, no se ha comprobado que esta sea la causa o la única causa, aunque todos los datos recogidos apuntan a ello).

Como ya se ha concluido, la opción utilizada a la hora de adquirir datos con el sensor LiDAR es la petición continua de datos con el comando HD y, posteriormente, esperar los datos de IMU y GNSS-RTK. También se ha comentado que hay un tiempo de espera para la petición de un nuevo dato al LiDAR, lo cual abre dos opciones:

- Mientras se espera el tiempo necesario para evitar el bloqueo del dispositivo, adquirir datos de los otros dos sensores tal como llegan, lo cual nos dejará varios datos de IMU y GNSS-RTK entre los datos del LiDAR.
- Tan solo recoger un dato de cada sensor y volver a esperar un dato LiDAR.

La diferencia entre ambas opciones es que con la primera de ellas se tendrían más datos a la hora de hacer medias de posiciones o inclinaciones en el posprocesamiento, con lo que no se depende de un solo dato para posicionar el sensor, pero los tiempos de recepción son más irregulares; mientras que con la segunda opción tan solo se trabaja con un solo dato de cada sensor pero los tiempos de recepción son bastante exactos si el LiDAR tiene el tiempo suficiente para procesar los datos.

Por el momento, ya que en el posprocesamiento no se realizan medias ni cálculos estadísticos de posicionamiento y lo que interesa para las siguientes versiones del código es aumentar en lo posible la rapidez sin llegar a bloquear, se elige la segunda opción de configuración ya que nos mostrará cuando el microcontrolador empieza a perder datos al no poder absorberlos todos y con eso se podrá apurar al máximo la rapidez de petición de datos. Y, lo más interesante de esta opción, es que es posible que no necesite poner un tiempo de espera para el envío de dato ya que el tiempo de espera al que le obligan los otros dos sensores sea suficiente, con lo que, por el momento, se ve la opción con mejores prestaciones.



2.2.2. Preprocesamiento

En este apartado se tratará lo que respecta al procesamiento de los datos adquiridos antes de ser guardados, las acciones realizadas para mejorar tiempos, la consecuencia de no reducirlos lo suficiente y las posibles soluciones a los problemas que ésto causa.

Receptor GNSS-RTK

El receptor GNSS-RTK es el sensor más sencillo de los tres a la hora de realizar el preprocesamiento, ya que una vez adquirido el dato, tan solo se le aplicará un filtro para seleccionar los datos que interesen, descartando el resto.

Para esto, se crea la función `"void USART_GPS_Filter_Data(char *text)"` con la que, definiendo con anterioridad los campos deseados, te devuelva éstos pasando por alto los campos no requeridos. Para dar más velocidad al proceso, lo que acaba devolviendo es el inicio de la dirección de memoria donde ha guardado los datos deseados, así se evitan transferencias de datos innecesarias.

El resto de procesos ya se realizaban en la primera parte del proyecto.

En cuanto al estudio de evitar el preprocesamiento y guardar los datos directamente en bruto, se concluye que al ser más bien un filtrado que un preprocesamiento, se está ganando tiempo en vez de perdiéndolo realizando este preprocesamiento, con lo que se decide mantenerlo activo.

Sensor IMU

El cambio realizado en el preprocesamiento del sensor inercial es trabajar sin números decimales, es decir, una vez realizado el proceso de obtención de los valores de Roll, Pitch y Yaw; sabiendo que éstos vienen con una precisión de dos decimales, lo que se hace es una multiplicación de ellos por cien para evitar los números decimales. Esto evita una serie de errores posteriores que se daban a la hora de realizar el posprocesamiento.

En cuanto al estudio de evitar el preprocesamiento y guardar los datos directamente en bruto, se concluye que no merece la pena al ser mínimo el tiempo que ahorraría en comparación con el volumen de trabajo que esto genera tanto en el posterior guardado como en el



posprocesamiento, con lo que se seguirá realizando el procesamiento de los datos del sensor antes de su guardado.

Sensor LiDAR

En cuanto al sensor LiDAR, el preprocesamiento de los datos ya estaba bien definido en el proyecto anterior, con lo que en éste tan solo se ha realizado el estudio para ver si es viable guardar los datos sin procesarlos previamente.

Se han seguido diferentes pasos para comprobarlo, el primero de todos ha sido comprobar la forma en la que afecta al tiempo de procesado el bloque preprocesamiento-guardado, y éste concluye en que el tiempo de éstos es bastante significativo, entre los veinticinco y los treinta milisegundos de media. La segunda parte, que es evitar el preprocesamiento y guardar todos los datos en bruto nos deja unos datos en los que marca que el tiempo de guardado está por debajo de los diez milisegundos, lo que hace que se ahorre alrededor de un 70% del tiempo utilizado, pero el guardado de los datos es bastante caótico, y el posprocesamiento con R y Matlab tal como están preparados no es viable.

Para ponerle solución a este problema, se ha intentado realizar un código para la misma placa que lo que realizase fuese la lectura de la memoria SD y el posterior procesado para dejar los datos tal y como los deja actualmente el código. Por problemas de tiempo no es posible la realización de este código, ya que es más complejo de lo que en un principio parecía, y se abre la puerta de o bien adaptarse al protocolo actual y conseguir hacerlo o, si éste es demasiado complicado, estudiar si un nuevo protocolo podría hacer las cosas más fáciles.

Como conclusión, por el momento se mantiene el preprocesamiento en la placa de adquisición, ya que a pesar de que sin él se podría ahorrar mucho tiempo, la temporalidad del proyecto no permite añadir un nuevo apartado en el cual trabajar este preprocesamiento después de adquirir todos los datos, y el objeto más importante del proyecto actual es la reconstrucción de la nube de puntos y no tanto la gestión temporal de la adquisición de datos.



2.2.3. Almacenamiento

En este apartado se tratará lo que respecta al almacenamiento de los datos preprocesados, la forma en la que se guardan para que los procesos posteriores, ya en ordenador, sean posibles, así como la mejora del código para intentar evitar bloqueos que se creen debidos a la tarjeta de guardado.

En primer lugar, el guardado de datos debe tener una estructura clara y fija, ya que posteriormente se deberá utilizar estos datos para su trabajo y éste debe ser automático y rápido. Para eso, lo primero que se guarda en la memoria SD son una fila de tres “-1”, para así marcar que la matriz será de anchura 3. A partir de allí, la siguiente lectura ya será de uno de los sensores, tal y como está configurado, será una lectura del sensor LiDAR.

El guardado del sensor LiDAR, en el caso de haber preprocesado los datos recibidos, es de la siguiente forma:

- Guarda su marcador de inicio de dato, que es “-18200”.
- En la misma fila, tras una tabulación, se guarda el tiempo actual del Timer 6 del microcontrolador (en milisegundos).
- En la figura 9 se muestra, en la primera línea, el ejemplo de guardado de las dos descripciones anteriores.

-18200	75902	-1
60000	-1	-1
60000	-1	-1
60000	-1	-1
60000	-1	-1
60000	-1	-1

Figura 9: Guardado de un dato LiDAR, inicio de la muestra.

- En la siguiente fila se guarda la distancia del primer ángulo, en caso de haber segundo y tercer retorno, se guardan en la misma línea añadiendo tabulaciones.
- En las siguientes filas se guardan los 1080 datos restantes (con o sin multiretorno de la misma forma que el primero).



- En la figura 10 se muestra cómo se guardan los datos en caso de que exista un multiretorno.

2976	-1	-1
2617	3244	-1
2685	-1	-1
2693	-1	-1
2566	3609	-1
3541	-1	-1

Figura 10: Guardado de un dato LiDAR, aparición de un multiretorno

- Por último, en la última fila, se guarda el marcador de terminación de dato "-18300" y, tras una tabulación, el tiempo en el que termina el guardado, tal y como se muestra la figura 11.

60000	-1	-1
60000	-1	-1
60000	-1	-1
60000	-1	-1
60000	-1	-1
-18300	75942	-1

Figura 11: Guardado de un dato LiDAR, final de muestra

Como se ha comentado, se busca la rapidez del proceso, y tanto el proceso de preprocesamiento de los datos LiDAR como el proceso de guardado son pesados.

El de guardado se intenta solucionar con la concatenación de los datos y guardarlos todos a la vez para aprovechar el almacenado de bloques de 512 bytes (tamaño del sector) que tiene el sistema de ficheros FAT utilizado. En este punto ha aparecido un problema de memoria RAM el cual, por el momento, no se ha solventado del todo, y tan solo se consiguen concatenar algunas partes, con lo que se decide seguir guardando los datos como hasta ahora y dejar el proceso de concatenación para más adelante, ya que si se evita el preprocesamiento, no sería necesaria la concatenación, y evitar el preprocesamiento es prioritario en futuras versiones.

En caso de acabar realizando el preprocesamiento de los datos posteriormente a la adquisición de éstos, el proceso de guardado sería sencillo, tan solo se colocaría la marca de principio con su tiempo, posteriormente se guardaría toda la cadena de datos LiDAR y por último la marca de final de dato y su tiempo, con lo que se evitaría la entrada a memoria demasiadas veces, además de ahorrar todo el tiempo que el preprocesamiento y el guardado conllevan.



Tras guardar los datos del LiDAR, se guarda un dato de la unidad de medida inercial, cuya estructura de guardado es la siguiente:

- Guarda su marcador de inicio de dato, que es “-18500”.
- En la misma fila, tras una tabulación, se guarda el tiempo actual del Timer 6 (en milisegundos).
- En la siguiente fila se guarda en la primera columna el ángulo Row, en la segunda el ángulo Pitch y en la tercera el ángulo Yaw.
- En la figura 12 se muestra un ejemplo del guardado de este sensor.

```
-18500  75942  -1
-605    -265   -12863
```

Figura 12: Guardado de dato IMU

Al igual que el LiDAR, se estudia la concatenación pero, al ser tan solo tres veces las que guarda, no se le da importancia y en la última versión del programa se guardan las cosas de una en una, aunque no se descarta volver a guardar todo de golpe una vez concatenadas todas las partes. Por el momento se evita ya que en alguna ocasión había dado problemas.

Por último se guarda un dato del receptor GNSS-RTK, cuya estructura de guardado es la siguiente:

- Guarda su marcador de inicio de dato, que es “-18400”.
- En la misma fila, tras una tabulación, se guarda el tiempo actual del Timer 6 (en milisegundos).
- En la siguiente fila guarda en la primera columna la posición UTM **X**, en la segunda columna la posición UTM **Y**, y en la tercera columna la altura sobre el nivel del mar **Z**.
- En la figura 13 se muestra un ejemplo del guardado de este dispositivo.

```
-18400  75986  -1
299716.357  4611430.192  190.766
```

Figura 13: Guardado de dato GNSS-RTK

Al igual que con el sensor IMU, para evitar problemas y ya que prácticamente no tienen ningún efecto, se evita la concatenación de los datos para su posterior guardado y se decidió guardar cada cosa en una orden de escritura.



2.2.4. Señalización y botones de usuario

Es importante poder saber, en cada momento, el estado en el que se encuentra el proceso, sobre todo cuando se está cambiando constantemente las versiones del código para mejorar uno u otro aspecto.

Se tiene la posibilidad de enviar mensajes de texto a través del USART al ordenador, pero esa posibilidad se convierte en un problema cuando las pruebas comienzan a ser con un sistema portátil. También se tiene la posibilidad de realizar la compra de un LCD y enviar allí los mensajes pertinentes.

Pero estas dos opciones pueden ser insuficientes cuando el problema es un bloqueo que no nos permite volver a pasar por la línea de código que enviaría el mensaje, con lo cual se piensa en un sistema que utiliza los recursos mínimos, y que, en caso de bloqueo, también nos señalice de alguna forma que el programa está bloqueado. El sistema de señalización será la utilización de los 4 LEDs de usuario que implementa la placa electrónica utilizada.

Hay que tener en cuenta que esta señalización no tiene por qué ser la definitiva, se pueden quitar, añadir o cambiar las señales según los intereses del usuario, es más, con el cambio de versiones e intereses de visualización en cada momento, se han ido cambiando estas señales según interesaba.

Se tienen 7 tipos de programas diferentes, y para cada uno de ellos se tiene la señalización pertinente, las cuales serán descritas a continuación.

Señalizaciones LED:



LED Encendido



LED Parpadeando



LED Apagado



LED Bloqueado (Encendido o Apagado pero sin cambiar de estado)



Tres sensores simultáneos, LiDAR HD:

	Programa inicializado, SysTick y Timer6 configurados.
	Configurando PushButton, USART1, USART2, USART6 y GPIO SD.
	No se detecta tarjeta SD.
	Error inicializando tarjeta SD.
	Error montando directorio en SD.
	Error creando archivo en directorio SD.
	Configurado PushButton, USART 1, 2, 6 y GPIO SD; Inicializada SD.
	Inicializando USART1 y USART6.
	Inicializado USART1 y USART6.
	Esperando posición precisa del GNSS-RTK.
	Posición precisa GNSS-RTK recibida.
	Estableciendo conexión con dispositivo LiDAR.
	Conexión con dispositivo LiDAR establecida.
	Encendiendo laser.
	Error LiDAR, el LiDAR ha enviado mensaje de error.
	Programa principal recibiendo datos.
	Programa principal en funcionamiento sin recibir datos de alguno de los sensores.
	Programa principal bloqueado.
	Programa principal finalizado, se puede extraer SD.
	Apagando laser.
	Laser apagado
	Desconectando LiDAR
	LiDAR Desconectado, todos los procesos han finalizado con éxito.



Tres sensores simultáneos, LiDAR ND:

	Programa inicializado, SysTick y Timer6 configurados. Configurando PushButton, USART1, USART2, USART6 y GPIO SD.
	No se detecta tarjeta SD.
	Error inicializando tarjeta SD.
	Error montando directorio en SD.
	Error creando archivo en directorio SD.
	Configurado PushButton, USART 1, 2, 6 y GPIO SD; Inicializada SD. Inicializando USART1 y USART6.
	Inicializado USART1 y USART6. Esperando posición precisa del GNSS-RTK.
	Posición precisa GNSS-RTK recibida. Estableciendo conexión con dispositivo LiDAR.
	Conexión con dispositivo LiDAR establecida. Encendiendo laser e inicializando comunicación continua ND.
	Error LiDAR, el LiDAR ha enviado mensaje de error.
	Programa principal recibiendo datos.
	Programa principal en funcionamiento sin recibir datos correctos del LiDAR.
	Programa principal bloqueado.
	Programa principal finalizado, se puede extraer SD. Apagando laser.
	Laser apagado Desconectando LiDAR
	LiDAR Desconectado, todos los procesos han finalizado con éxito.



Sensor LiDAR HD:

	Programa inicializado, SysTick y Timer6 configurados. Configurando PushButton, USART2 y GPIO SD.
	Configurado PushButton, USART2 y GPIO SD. Iniciando SD.
	No se detecta tarjeta SD.
	Error inicializando tarjeta SD.
	Error montando directorio en SD.
	Error creando archivo en directorio SD.
	Configurado PushButton, USART 2 y GPIO SD; Inicializada SD. Estableciendo conexión con dispositivo LiDAR.
	Conexión con dispositivo LiDAR establecida. Encendiendo laser.
	Error LiDAR, el LiDAR ha enviado mensaje de error.
	Programa principal recibiendo datos.
	Programa principal en funcionamiento sin recibir datos LiDAR.
	Programa principal bloqueado.
	Programa principal finalizado, se puede extraer SD. Apagando laser.
	Laser apagado Desconectando LiDAR
	LiDAR Desconectado, todos los procesos han finalizado con éxito.



Sensor LiDAR ND:



Programa inicializado, SysTick y Timer6 configurados.
Configurando PushButton, USART2 y GPIO SD.



Configurado PushButton, USART2 y GPIO SD.
Iniciando SD.



No se detecta tarjeta SD.



Error inicializando tarjeta SD.



Error montando directorio en SD.



Error creando archivo en directorio SD.



Configurado PushButton, USART 2 y GPIO SD; Inicializada SD.
Estableciendo conexión con dispositivo LiDAR.



Conexión con dispositivo LiDAR establecida.
Encendiendo laser.



Error LiDAR, el LiDAR ha enviado mensaje de error.



Encendido laser.
Iniciando comunicación continua ND.



Programa principal recibiendo datos.



Programa principal en funcionamiento sin recibir datos LiDAR



Programa principal bloqueado.



Programa principal finalizado, se puede extraer SD.
Apagando laser.



Laser apagado
Desconectando LiDAR



LiDAR Desconectado, todos los procesos han finalizado con éxito.



Sensores IMU y GNSS-RTK simultáneos:



Programa inicializado, SysTick y Timer6 configurados.
Configurando PushButton y GPIO SD.



Configurado PushButton y GPIO SD.
Iniciando SD.



No se detecta tarjeta SD.



Error inicializando tarjeta SD.



Error montando directorio en SD.



Error creando archivo en directorio SD.



Configurado PushButton y GPIO SD; Inicializada SD.
Configurando USART6.



Configurado USART6.
Configurando USART1.



Configurado USART1.
Esperando posición GNSS-RTK correcta.



Programa principal recibiendo datos



Programa principal finalizado, se puede extraer SD.



Sensor IMU:



Programa inicializado, SysTick y Timer6 configurados.
Configurando PushButton y GPIO SD.



Configurado PushButton y GPIO SD.
Iniciando SD



No se detecta tarjeta SD.



Error inicializando tarjeta SD.



Error montando directorio en SD.



Error creando archivo en directorio SD.



Configurado PushButton y GPIO SD; Inicializada SD.
Configurando USART6.



Programa principal recibiendo datos.



Programa principal finalizado, se puede extraer SD.



Sensor GNSS-RTK:



Programa inicializado, SysTick y Timer6 configurados.
Configurando PushButton y GPIO SD.



Configurado PushButton y GPIO SD.
Iniciando SD



No se detecta tarjeta SD.



Error inicializando tarjeta SD.



Error montando directorio en SD.



Error creando archivo en directorio SD.



Configurado PushButton y GPIO SD; Inicializada SD.
Configurando USART1.



Configurado USART1.
Esperando posición GNSS-RTK correcta.



Programa principal recibiendo datos.



Programa principal finalizado, se puede extraer SD.

Pulsadores:

La función de los pulsadores es la misma para todos los programas.

- **RESET (negro):** Reinicia el software forzosamente (es como quitar la corriente y volverla a poner).
- **PUSHBUTTON (azul):** Finaliza la captura de datos (Solo una vez dentro del programa principal).



2.3. Posprocesamiento de datos: Generación de la nube de puntos

Una vez se tienen los datos guardados en la SD, de forma correcta, hay que convertirlos en datos útiles, datos con los que se pueda trabajar o sacar conclusiones.

Para esto se utilizará un script realizado con el programa Matlab, el cual permite tratar todos los datos y no solo reconstruir la nube de puntos, sino también hacer diferentes análisis que ayudarán a la hora de visualizar cosas como los problemas temporales en la adquisición o verificar si los datos obtenidos por cada sensor son o no lógicos.

2.3.1. Preparación de los datos

Una vez se tiene el archivo de texto en el ordenador, se deben interpretar los datos que tiene dentro para posteriormente poder realizar los análisis pertinentes.

El primer paso de todos se realizará con el programa R Studio, un software libre encarado a la estadística, el cual permite adaptar la matriz de datos previamente al uso de ésta con el programa Matlab. La razón de este paso intermedio es que Matlab no acepta matrices con diferentes dimensiones entre sus vectores, y con el programa R Studio lo que se realiza es un relleno de los huecos vacíos por valores “-1”, para obtener como salida una matriz de dimensiones constantes.

Los motivos por los cuales sale una matriz en el guardado con dimensiones diferentes entre sus vectores son que depende del número de retornos recibidos por el sensor LiDAR y que en los vectores donde se guarda el identificador y el tiempo son de dimensión 2 en vez de 3.

El proceso con el programa R Studio abre este archivo de texto, reconoce que es una matriz de 3xZ ya que en la primera línea de guardado se ha grabado un vector de dimensión 3 con los valores “-1”, busca desde el inicio hasta la fila Z huecos vacíos y, en caso de hallarlos, guarda un “-1” en ellos. Una vez finalizado, genera un archivo de texto nuevo con el nombre del archivo introducido seguido por un “_to_Matlab”.

En este momento ya se dispone de un archivo de texto preparado para el trabajo con Matlab.

El primer paso es reconocer los identificadores, ya que son la clave para saber qué dato viene a continuación; estos identificadores son:



- Identificador de comienzo de guardado de dato LiDAR (L = -18200)
- Identificador de finalización de guardado de dato LiDAR (T = -18300)
- Identificador de comienzo de guardado de dato GNSS-RTK (G = -18400)
- Identificador de comienzo de guardado de dato IMU (I = -18500)

Una vez se detecta el identificador, y conociendo el formato de guardado de cada uno de los sensores, se procede a extraer los datos y clasificarlos en sus vectores correspondientes, ya sean tiempo, distancia, ángulo o posición.

Los datos del LiDAR son filtrados para que todo aquel dato mayor o igual que 60000mm o menor que 50mm desaparezca, para así evitar dibujar las medidas sin retorno y los retornos que puedan proceder de algún cable que se cruce un instante y pueda ensuciar la nube.

Los datos del receptor GNSS-RTK son pasados a milímetros, ya que se obtienen en metros, para así hacerlos coincidir con los obtenidos del sensor LiDAR, y posteriormente se les resta el valor de la coordenada UTM en milímetros del lugar donde se realice el escaneo, para que los valores con los que se trabaja sean menores y el script posteriormente trabaje más rápido al generar la nube de puntos. También se le resta la altura.

Todas las pruebas se realizan en el edificio 4 de ETSEA, con lo que los valores restados por el script de Matlab son los siguientes:

Valor restado en el eje X: 299765620.

Valor restado en el eje Y: 4611438623.

Valor restado en el eje Z: 190300.

El dato del ángulo Yaw del sensor inercial también debe ser modificado, ya que existe lo que se llama una declinación magnética, que es la diferencia en grados entre el Norte Geográfico y el Norte Magnético, como se muestra en la figura 14. Esta declinación es Este (declinación en signo positivo) u Oeste (declinación en signo negativo) dependiendo de dónde nos encontremos, cada lugar de la Tierra tiene una declinación distinta, en algunos casos es tan pequeña que no se tiene en consideración.

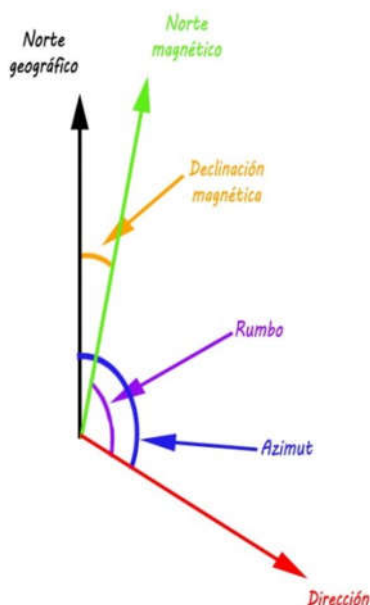


Figura 14: Diagrama vectorial de la declinación magnética

Por el momento no se ha encontrado una fuente clara a la hora de calcular la declinación, ya que hay autores con los que la declinación aparece prácticamente nula y para otros es de un ángulo ya considerable.

Para solucionar el problema, se hará un ensayo en el cual el dispositivo realice un recorrido totalmente recto y evitando cualquier inclinación, y se contrastará directamente en los datos el ángulo que marcarían los datos GNSS-RTK con el ángulo Yaw dado por el sensor inercial.

Con estos datos guardados ya en vectores de forma correcta, ordenada y filtrada, se puede proceder a hacer el programa funcional y útil para el usuario.

Hay que tener en cuenta que el script de Matlab comienza a coger los datos por el final del fichero, ya que así coge primero la información de los datos de los sensores GNSS-RTK e IMU, y después coge los datos del LiDAR y los manda a la función que genera la nube de puntos, la cual se explica en uno de los apartados posteriores.

Entonces, para poder tener unos datos coherentes para realizar los análisis, una vez leídos los datos, hay que girar las matrices para que el inicio de los datos coincida con el inicio de las medidas adquiridas, ya que de otra forma se verían los datos volteados.

2.3.2. Programa de apoyo al análisis

Una vez se tienen todos los datos en sus respectivos vectores de trabajo, se deben expresar al máximo para que sean lo más útiles posibles, no solo deben centrarse en la nube de puntos, que es su objetivo final, sino que deben servir para mejorar las versiones de los programas de adquisición de datos.

Para la realización de esta herramienta, lo primero que se hace es dejar vacío el menú de Matlab y, a partir de allí, generar un menú que permita realizar las acciones que sean necesarias.

En la figura 15 se puede ver cómo sería una figura predeterminada de Matlab, pudiéndola comparar con la figura 16, donde se ve el menú y estilo personalizados de la figura diseñada para este proyecto.

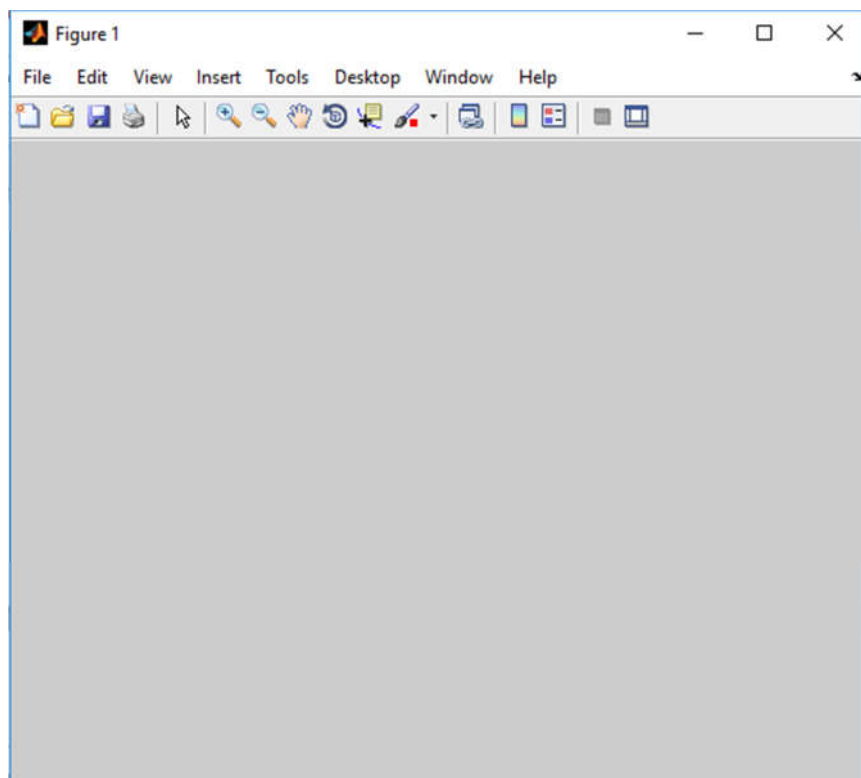


Figura 15: Figure estándar generado por Matlab

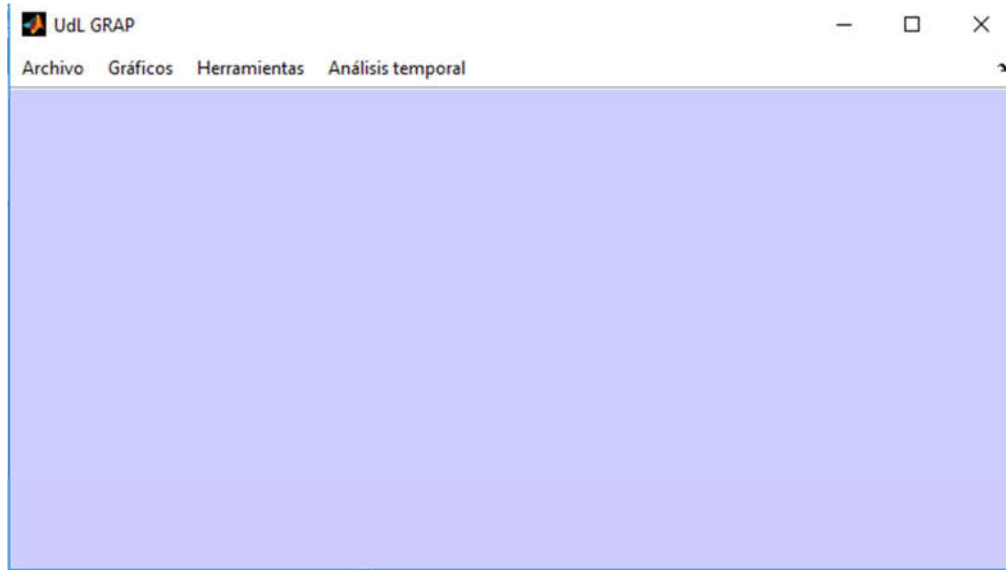


Figura 16: Figure Matlab adaptado

Evidentemente este menú puede ser modificado, se pueden añadir opciones, se pueden quitar, se puede reorganizar, está totalmente abierto a la personalización deseada por el usuario.

Más adelante se dará una explicación más extensa de las opciones del menú, pero antes hay que destacar otras funcionalidades de éste, como pueden ser:

- El programa se adapta a cualquier tamaño de pantalla, ya que mide el tamaño de esta y la parte en cuatro, haciendo cada pestaña del tamaño de un cuarto de pantalla.
- Las pestañas se van abriendo en orden en la pantalla, es decir, de la 1 a la 4 y, si por ejemplo, están las tres primeras abiertas y la segunda es cerrada, al elegir la opción nueva pestaña, ya sea desde el menú de la pestaña 1 o de la 3, se abrirá la pestaña 2 y no la 4.
- Las pestañas se pueden poner en pantalla completa y, al volverlas a minimizar, volverán a ocupar el espacio que les correspondía al principio.
- Las gráficas adaptan automáticamente el valor de sus ejes para abarcar todos sus datos y a su vez que se vea con la máxima resolución posible.

Estas características, que podrían parecer de menor importancia, dan una comodidad especial a la hora de trabajar, y es por eso que se les da un valor añadido.



Respecto al menú, tiene varias características y opciones, las cuales se van a analizar una a una para que se comprenda cada una de las partes de éste y, en caso de que el usuario pueda echar en falta algo en el menú, lo transmita y pueda ser modificado en la siguiente versión del programa.

Se comenzará con la pestaña “**Archivo**”, la cual nos permite realizar acciones que podrían denominarse generales.

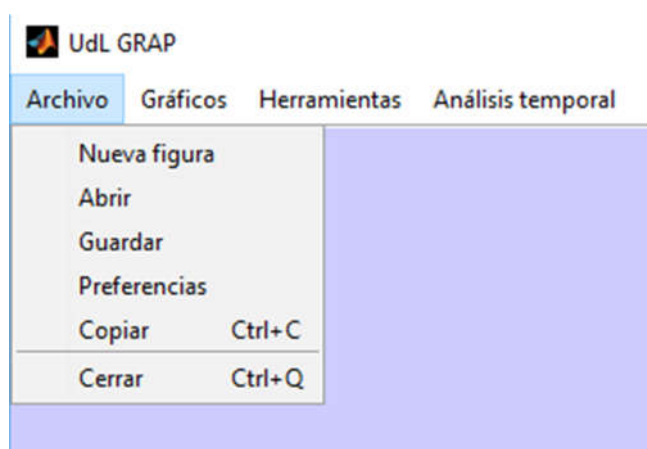


Figura 17: Menú Archivo

A continuación se expone cada una de las opciones permitidas dentro del menú Archivo que se muestra en la figura 17:

- Nueva figura:
 - Genera una nueva figura hasta un máximo de 4. Si ha llegado al máximo de 4 figura aparece el menú contextual que se muestra en la figura 18 avisando de que ya no se pueden abrir más figuras por haber llegado al máximo.

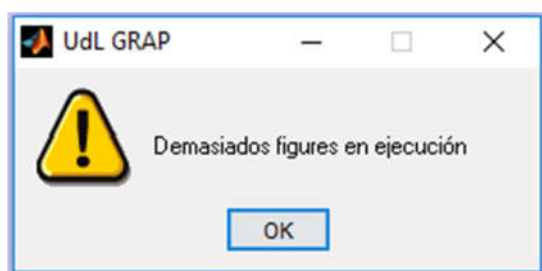


Figura 18: Mensaje de advertencia



- Las figuras son generadas siempre en orden, por ejemplo, si están abiertas las 3 primeras y se cierra la segunda, cuando pidamos una nueva figura se abrirá la segunda y no la cuarta. El orden de las pestañas se puede ver en la figura 19.
- El código se adapta a cualquier pantalla de ordenador, ya que calcula automáticamente el tamaño de la pantalla, lo divide en cuatro, y así genera el tamaño correspondiente para las 4 figuras.

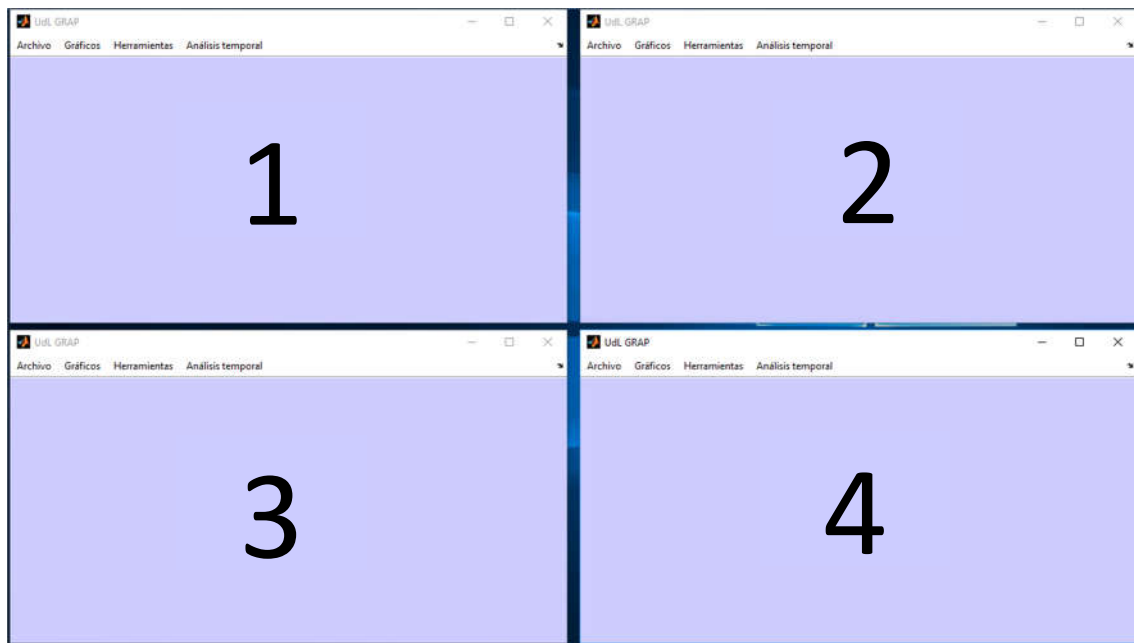


Figura 19: Orden de apertura de nuevas figuras

- **Abrir:**
Abre el sistema de directorios de Windows y te permite navegar por él y abrir el fichero de datos que se desea analizar o a partir del cual se quiere generar la nube de puntos.
- **Guardar:**
Una vez abierto el fichero, guarda dos archivos en la carpeta donde se encuentre el script de Matlab, uno guarda la nube de puntos y tiene el nombre de “Nube.txt” y el otro guarda el recorrido GNSS-RTK y tiene el nombre de “GPS.txt”.

- Preferencias:

Abre el menú de Matlab de configuración de preferencias mostrado en la figura 20.

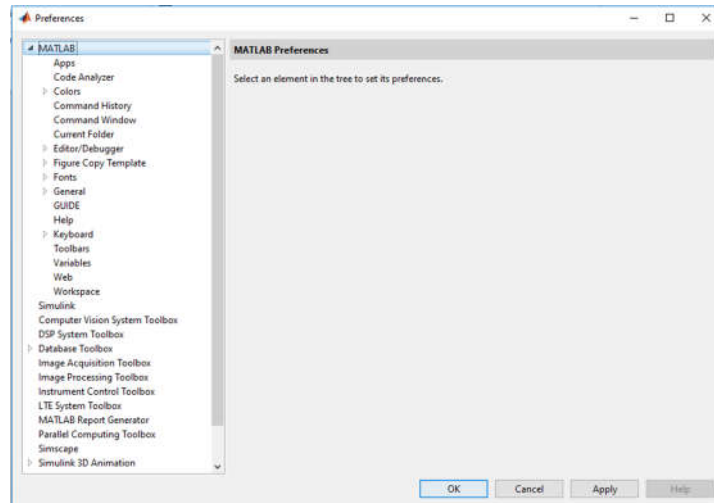


Figura 20: Menú de preferencia de Matlab

- Copiar:

Permite copiar el contenido actual del figure en el portapapeles para poder utilizarlo en lo que se desee, sobretodo encarado a la preparación de informes.

La siguiente pestaña es la de “Gráficos”, la cual nos permite ver los datos recogidos de cada uno de los sensores.

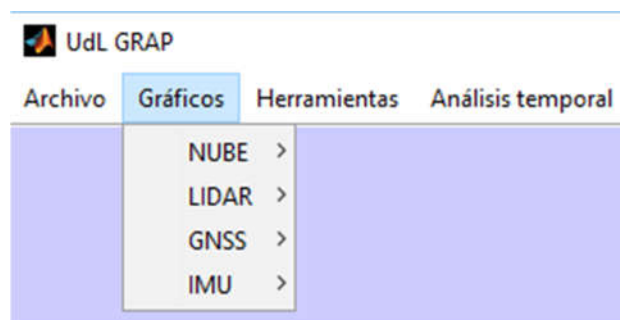


Figura 21: Menú Gráficos

A continuación se expone cada una de las opciones permitidas dentro del menú mostrado en la figura 21:

- NUBE:

Permite visualizar la nube de puntos generada desde diferentes puntos de vista. Las funciones de este apartado son para hacerse una idea de la nube generada y no para navegar por ella o ver las cosas con precisión, ya que se considera innecesario teniendo un programa de visualización 3D de nubes de puntos como puede ser CloudCompare.

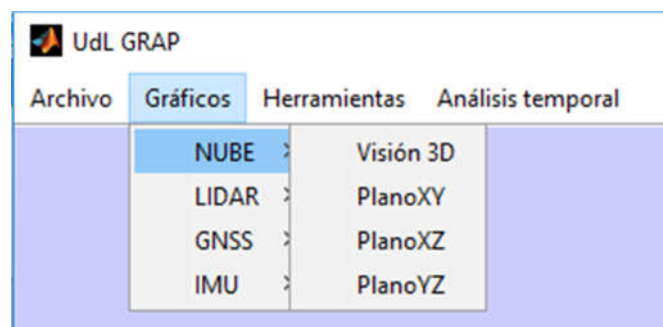


Figura 22: Menú Gráficos-NUBE

Los submenús que se pueden ver en la figura 22 dentro de NUBE son:

○ Visión 3D:

Como se muestra en la figura 23, ofrece una visualización en 3D de la nube generada.

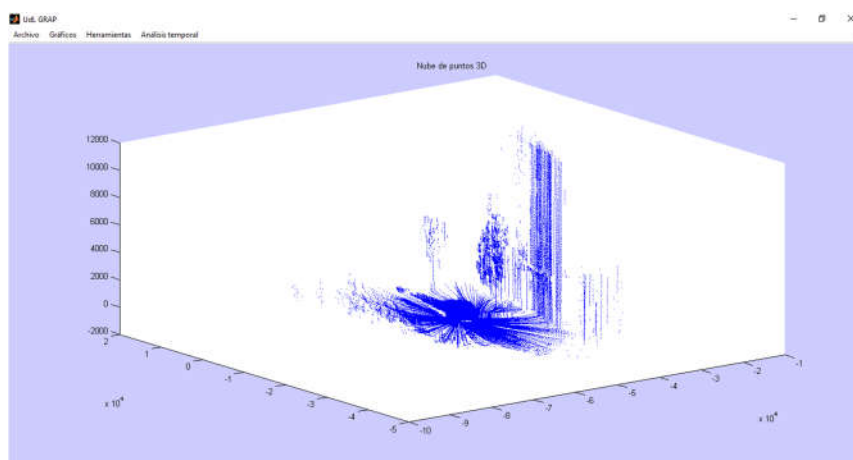


Figura 23: Nube de puntos, visión 3D

- Plano XY:

Como se muestra en la figura 24, ofrece una visualización del plano de corte XY.

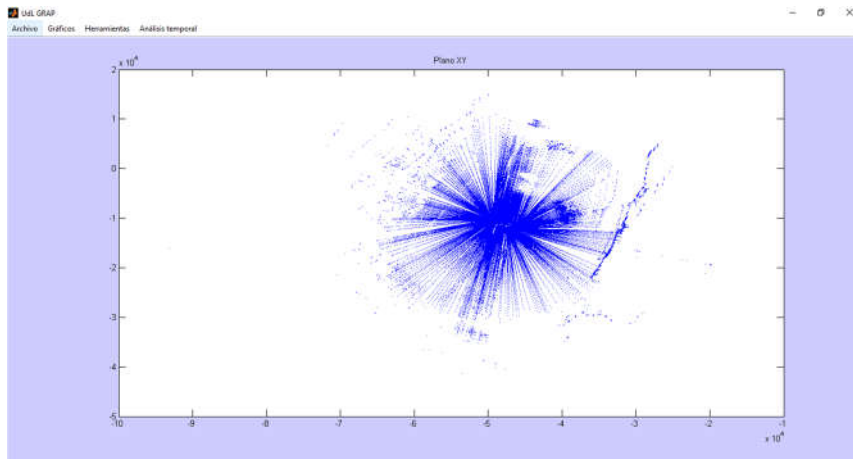


Figura 24: Nube de puntos, plano de corte XY

- Plano XZ:

Como se muestra en la figura 25, ofrece una visualización del plano de corte XZ.

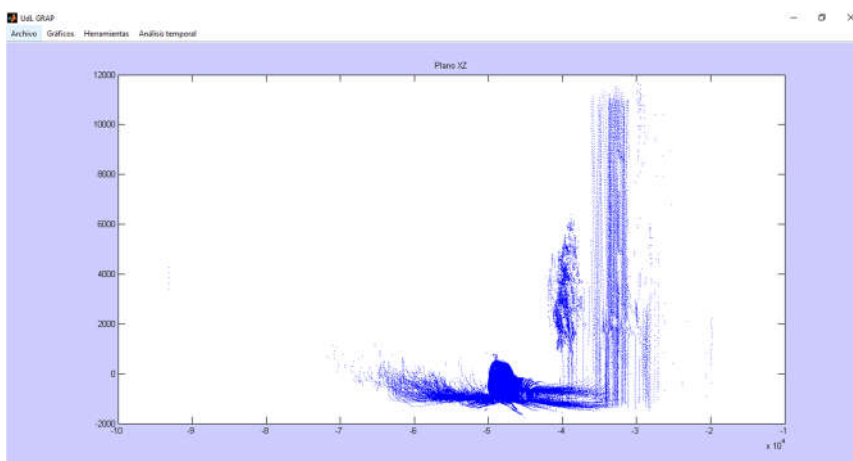


Figura 25: Nube de puntos, plano de corte XZ

- Plano YZ:

Como se muestra en la figura 26, ofrece una visualización del plano de corte YZ.

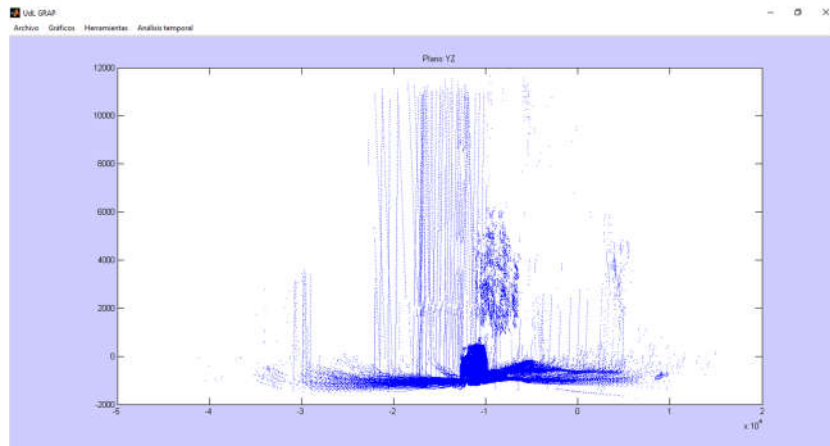


Figura 26: Nube de puntos, plano de corte YZ

- LiDAR:

Permite visualizar los datos recibidos del LiDAR, eligiendo el scan deseado y los retornos que se quieran ver, así como navegar de unos a otros cómodamente.

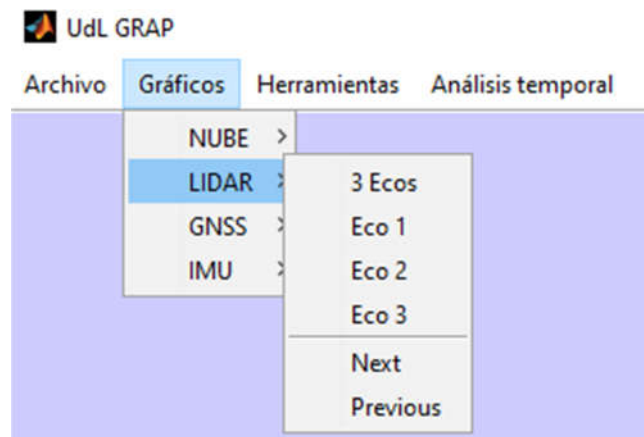
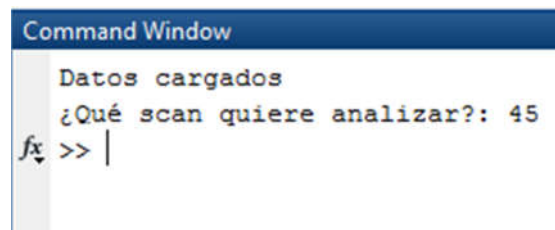


Figura 27: Menú Gráficos-LiDAR



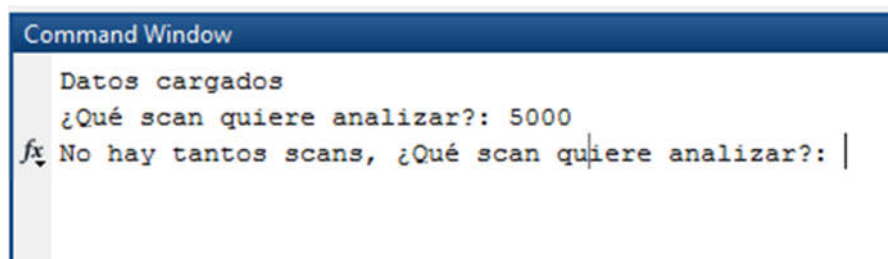
Para escoger el scan deseado, una vez escogido el retorno que se requiere, aparece en el Command Window de Matlab una petición de introducción del scan deseado, tal y como se muestra en la figura 28.



```
Command Window
Datos cargados
¿Qué scan quiere analizar?: 45
fx >> |
```

Figura 28: Selección del scan a analizar

En caso de que este no exista, el programa nos mostrará en el Command Window un mensaje de advertencia y nos permitirá escoger de nuevo el scan a analizar, como se puede observar en la figura 29.



```
Command Window
Datos cargados
¿Qué scan quiere analizar?: 5000
fx No hay tantos scans, ¿Qué scan quiere analizar?: |
```

Figura 29: Mensaje de advertencia, scan no alcanzable, no se han realizado tantos

Los diferentes submenús mostrados en la figura 27 para la pestaña LiDAR son:

- 3 Ecos:

Muestra los tres retornos del scan escogido, el primer retorno lo mostraría en color rojo, el segundo retorno en color verde y el tercer retorno en color azul. Como se puede observar en la figura 30, en el eje de ordenadas se muestra la distancia medida por el LiDAR y en el eje de abscisas el ángulo de giro del láser LiDAR.

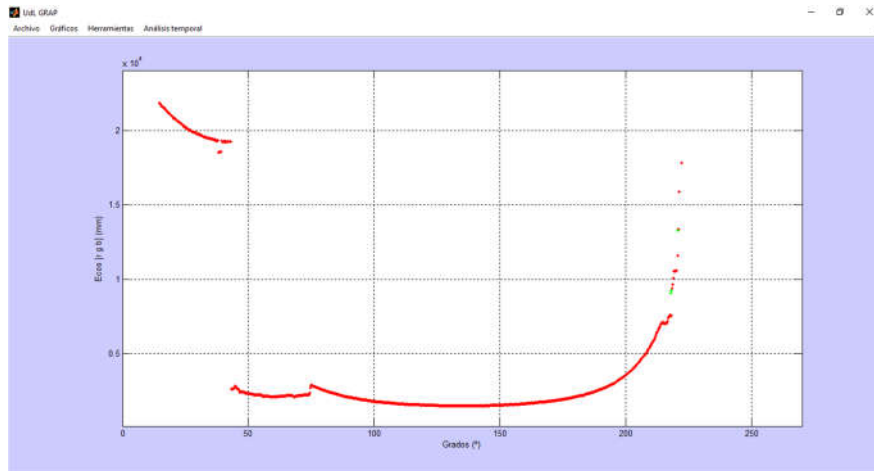


Figura 30: Graficado de todos los datos LiDAR recogidos

○ Eco 1:

Muestra el primer retorno del LiDAR en color rojo.

Como se puede observar en la figura 31, en el eje de ordenadas se muestra la distancia medida por el LiDAR y en el eje de abscisas el ángulo de giro del láser LiDAR.

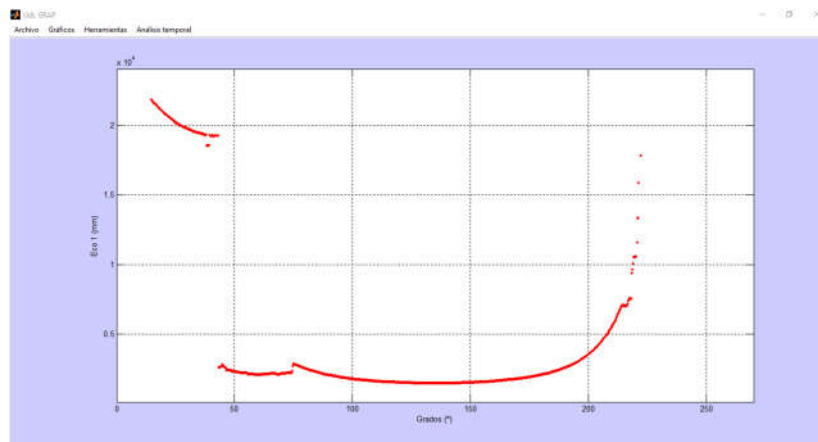


Figura 31: Graficado del primer rebote LiDAR



- Eco 2:

Muestra el segundo retorno del LiDAR en color verde.

Como se puede observar en la figura 32, en el eje de ordenadas se muestra la distancia medida por el LiDAR y en el eje de abscisas el ángulo de giro del láser LiDAR.

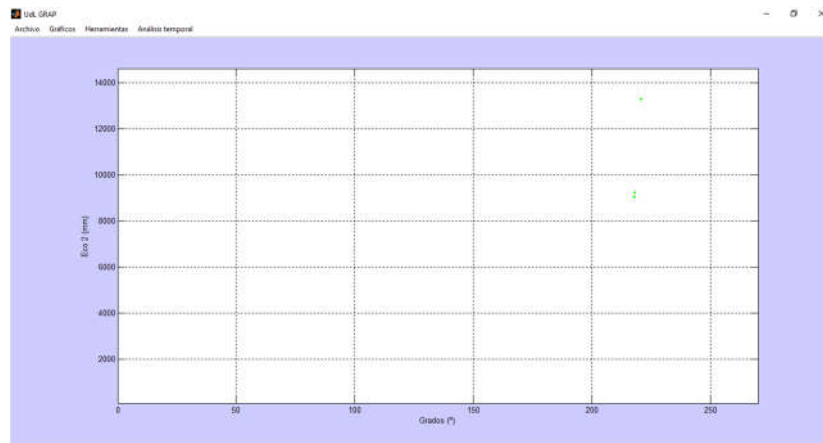


Figura 32: Graficado del segundo rebote LiDAR

- Eco 3:

No se ha encontrado ningún tercer retorno en este scan, si este existiese, saldría graficado en azul con el mismo formato que el resto de retornos.

Hay que tener en cuenta que al no existir uno de los retornos, Matlab muestra en su Command Window un mensaje de error.

El formato del gráfico es el mismo que en los casos anteriores, en el eje de ordenadas se muestra la distancia medida por el LiDAR y en el eje de abscisas el ángulo de giro del láser LiDAR.

- Next:

Una vez escogido el scan deseado, se puede ir al posterior sin necesidad de realizar todo el proceso de elección de nuevo.

- Previous:

Una vez escogido el scan deseado, se puede ir al anterior sin necesidad de realizar todo el proceso de elección de nuevo.

- GNSS:

Permite observar los datos obtenidos del receptor GNSS-RTK de forma cómoda y accesible.

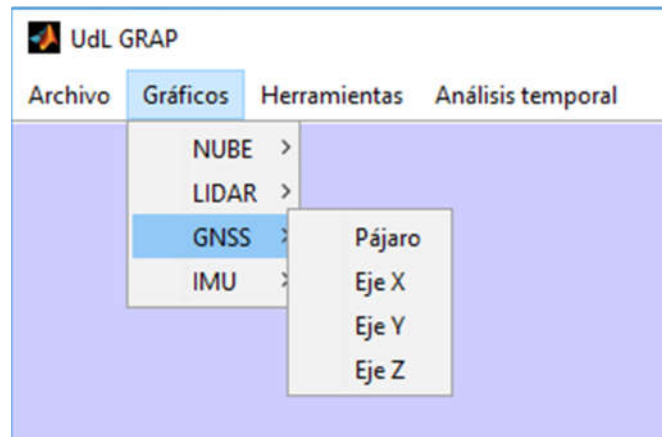


Figura 33: Menú Gráficos-GNSS

Como se puede observar en la figura 33, el menú GNSS cuenta con otros cuatro submenús:

○ Pájaro:

Permite hacer una visualización en vista de pájaro o aérea del recorrido realizado, es decir, como se muestra en la figura 34, muestra el recorrido haciendo un corte del plano XY.

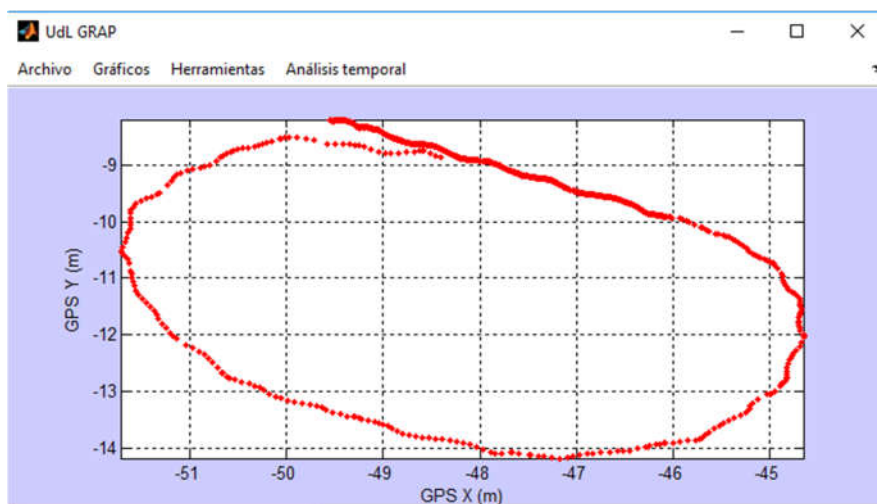


Figura 34: Graficado de los datos del receptor GNSS-RTK, corte del plano XY

- Eje X:

Como muestra la figura 35, permite ver la evolución de la coordenada X a lo largo del recorrido realizado.

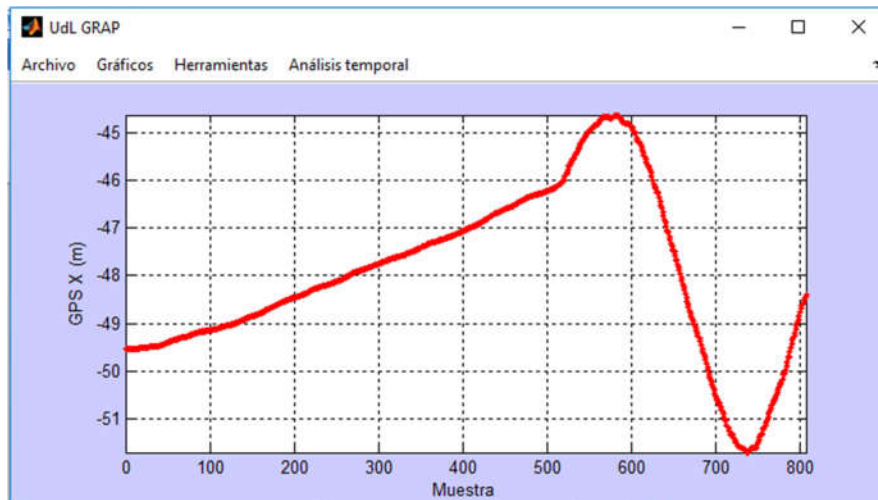


Figura 35: Graficado de la coordenada X del receptor GNSS-RTK

- Eje Y:

Como muestra la figura 36, permite ver la evolución de la coordenada Y a lo largo del recorrido realizado.

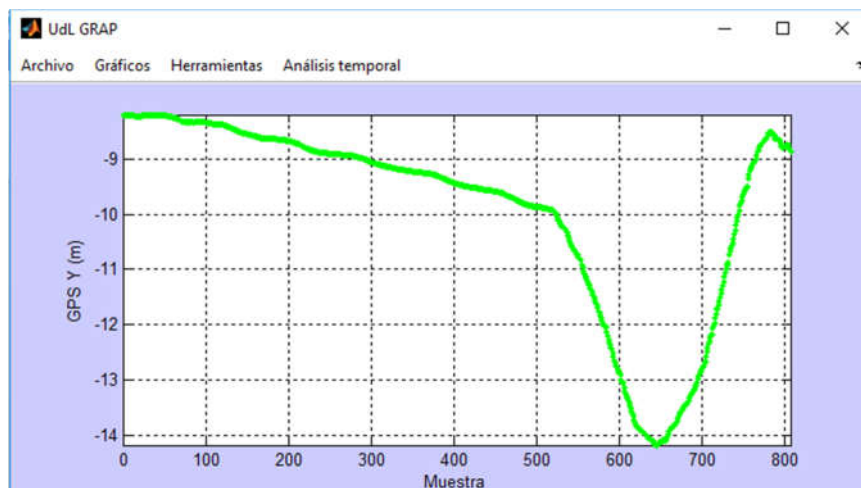


Figura 36: Graficado de la coordenada Y del receptor GNSS-RTK

- Eje Z:

Como muestra la figura 37, permite ver la evolución de la coordenada Z a lo largo del recorrido realizado.

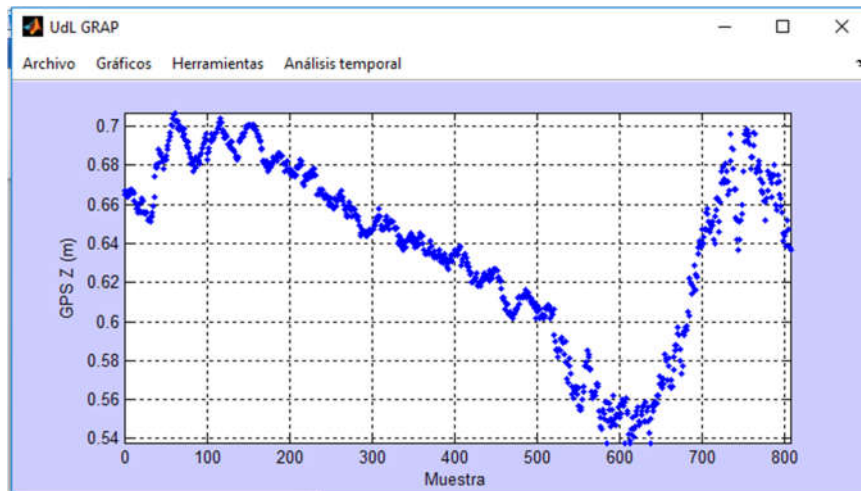


Figura 37: Gráfico de la coordenada Z del receptor GNSS-RTK

- IMU:

Permite visualizar los datos obtenidos por el sensor inercial y así poder evaluarlos de una forma visualmente sencilla.

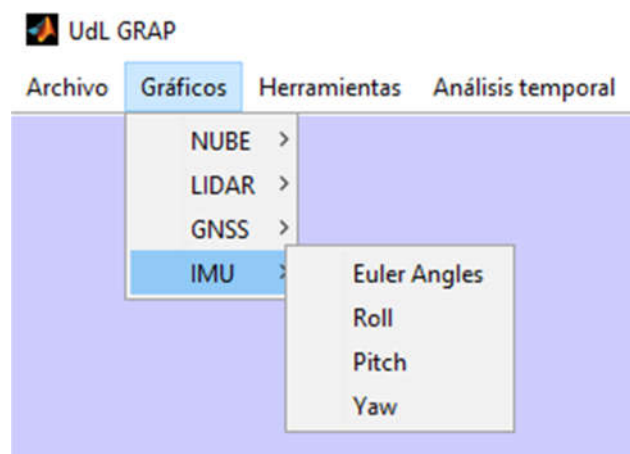


Figura 38: Menú Gráficos-IMU

Como se puede observar en la figura 38, el menú IMU cuenta con otros cuatro submenús:

- Euler Angles:

Muestra la evolución de los tres ángulos de Euler en grados.

Como se puede observar en la figura 39, el eje de ordenadas muestra los grados de apertura de cada ángulo y el eje de abscisas muestra el número de muestra analizada.

El ángulo Roll se muestra en color rojo, el ángulo Pitch se muestra en color verde y el ángulo Yaw se muestra en color azul.

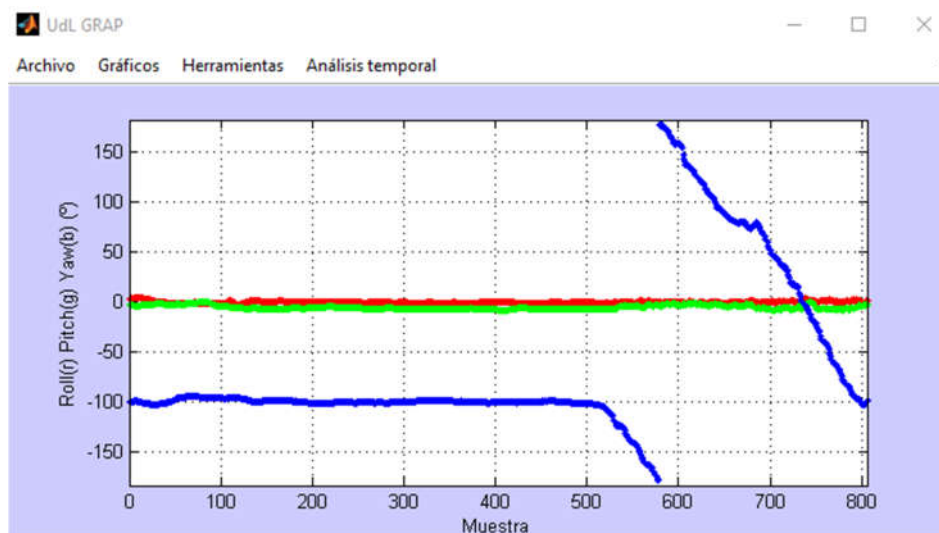


Figura 39: Graficado de los tres ángulos de la unidad de medida inercial

- Roll:

Muestra la evolución de la inclinación Roll en grados.

Como se puede observar en la figura 40, el eje de ordenadas muestra los grados de apertura del ángulo y el eje de abscisas muestra el número de muestra analizada.

Los datos se muestran en color rojo para que siga el mismo protocolo que el submenú EulerAngles.

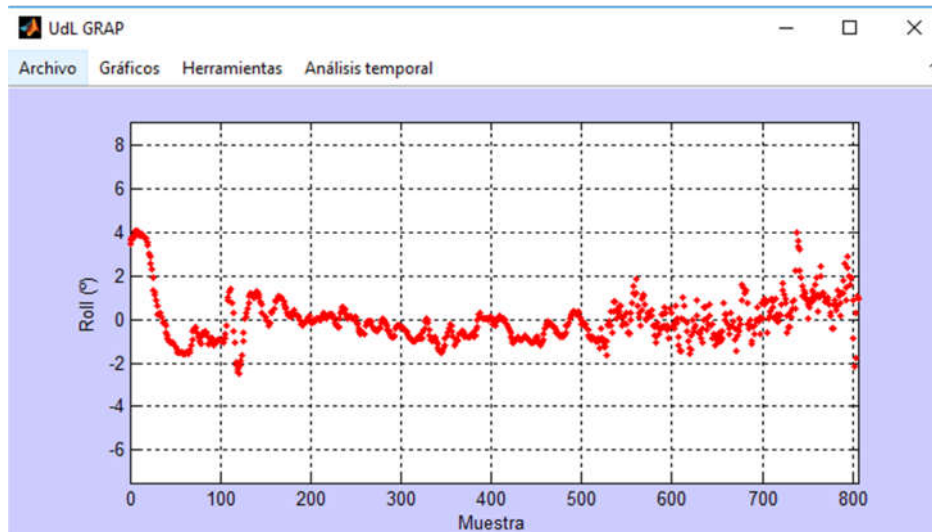


Figura 40: Graficado del ángulo roll de la unidad de medida inercial

○ Pitch:

Muestra la evolución de la inclinación Pitch en grados.

Como se puede observar en la figura 41, el eje de ordenadas muestra los grados de apertura del ángulo y el eje de abscisas muestra el número de muestra analizada.

Los datos se muestran en color verde para que siga el mismo protocolo que el submenú EulerAngles.

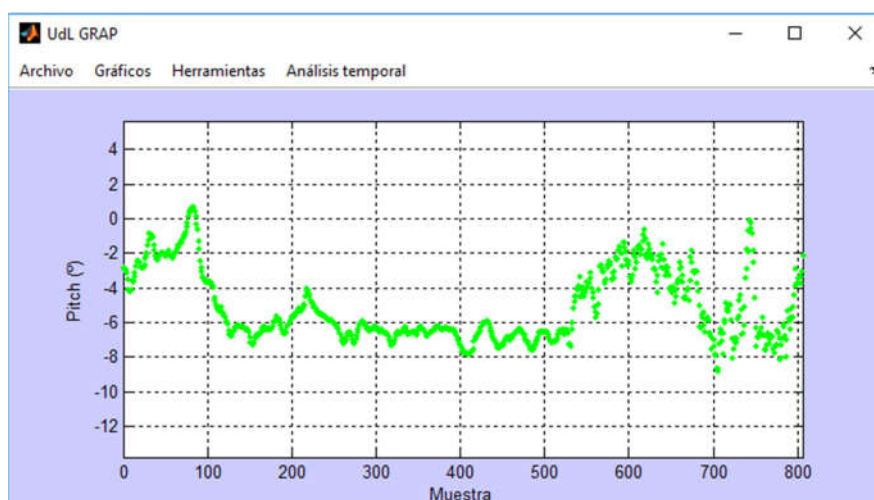


Figura 41: Graficado del ángulo pitch de la unidad de medida inercial

- Yaw:

Muestra la evolución de la inclinación Yaw en grados.

Como se puede observar en la figura 42, el eje de ordenadas muestra los grados de apertura del ángulo y el eje de abscisas muestra el número de muestra analizada.

Los datos se muestran en color azul para que siga el mismo protocolo que el submenú EulerAngles.

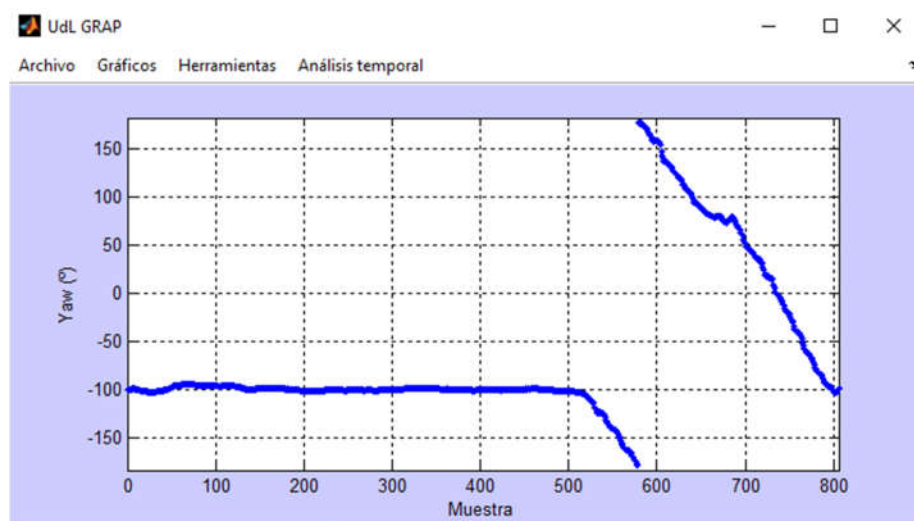


Figura 42: Gráficoado del ángulo yaw de la unidad de medida inercial

La tercera pestaña es la de “Herramientas”, la cual nos permite realizar acciones sobre la gráfica una vez cargada.

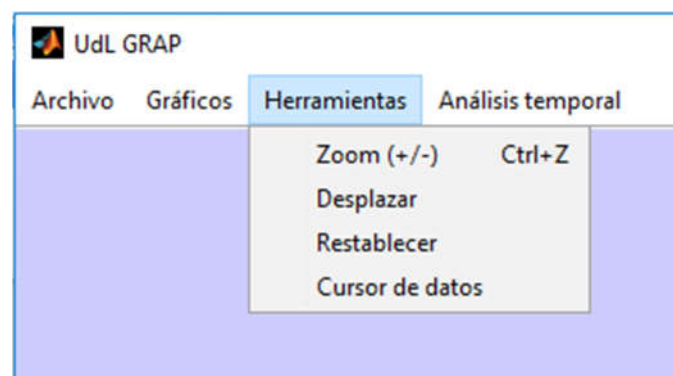


Figura 43: Menú Herramientas



Como se puede observar en la figura 43, el menú Herramientas cuenta con otros cuatro submenús:

- Zoom (+/-):

Una vez activo tiene las siguientes opciones:

- Botón izquierdo: Acerca el zoom, amplía la zona que se clicca.
- Botón derecho: Aleja el zoom, reduce la zona que se clicca.
- Selección: Adapta la visión al área seleccionada.
- Rosca hacia delante: Acerca el zoom.
- Rosca hacia detrás: Aleja el zoom

Para activarlo, se puede clicar esta opción del menú o simplemente se puede utilizar el atajo Ctrl+Z del teclado.

- Desplazar:

Una vez activo, permite desplazar la gráfica lateral o verticalmente arrastrando con el botón izquierdo del ratón pulsado.

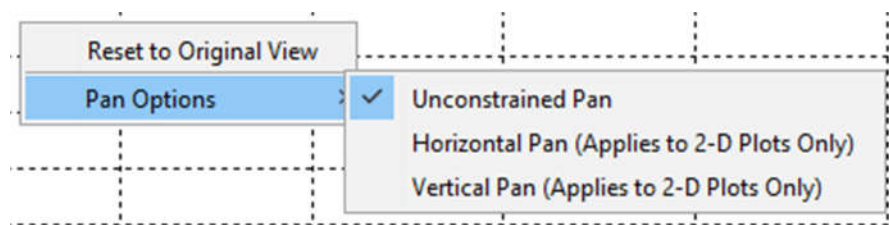


Figura 44: Menú de opciones de la herramienta desplazar

Como se muestra en la figura 44, si se clicca el botón derecho sobre el gráfico, aparece un menú contextual con las siguientes opciones:

- Reset to Original View:
Vuelve a la vista inicial del gráfico.
- Pan Options -> Unconstrained Pan
Si ésta es la opción seleccionada, permite desplazar el gráfico en todas las direcciones.
- Pan Options ->Horizontal Pan
Si ésta es la opción seleccionada, tan solo permite el desplazamiento horizontal del gráfico.

- Pan Options -> Vertical Pan

Si ésta es la opción seleccionada, tan solo permite el desplazamiento vertical del gráfico.

- Restablecer:

Vuelve a la vista inicial del gráfico.

- Cursor de datos:

Como se puede observar en la figura 45, nos permite conocer los valores exactos de cualquier punto dibujado en la gráfica. Este punto, una vez colocado, puede ser desplazado con facilidad tan solo arrastrándolo con el botón izquierdo.

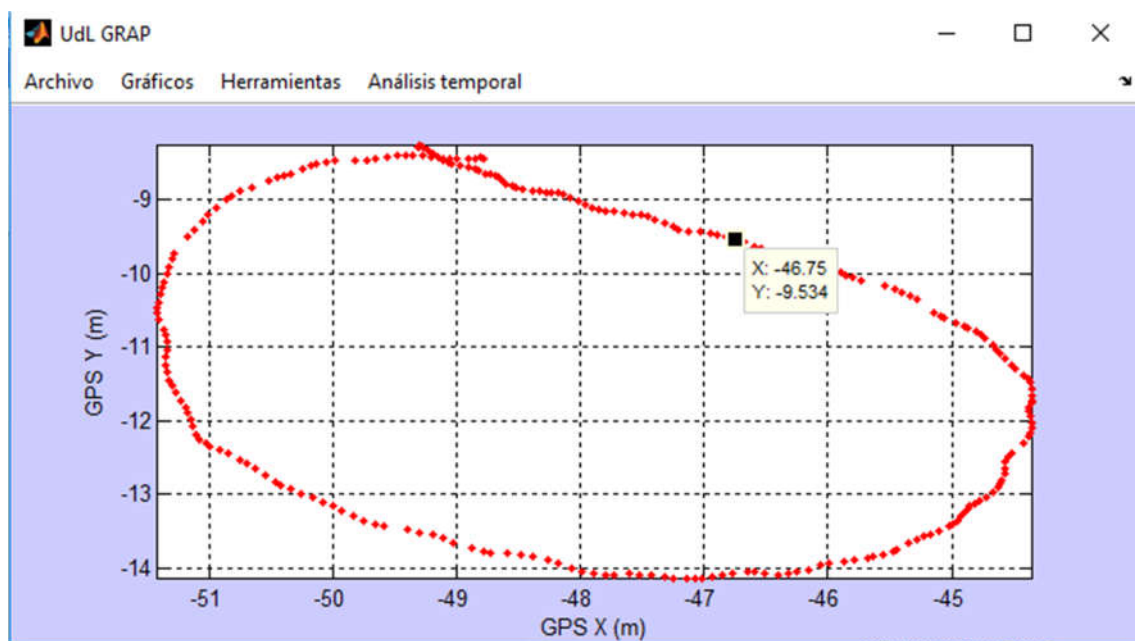


Figura 45: Visualización de datos con el cursor de datos

Una vez puesto el primer punto, si se desea compararlo con otros puntos, es posible seleccionar varios en la misma gráfica si cuando se clic se mantiene a la vez pulsado el botón "alt". El resultado es el mostrado en la figura 46.

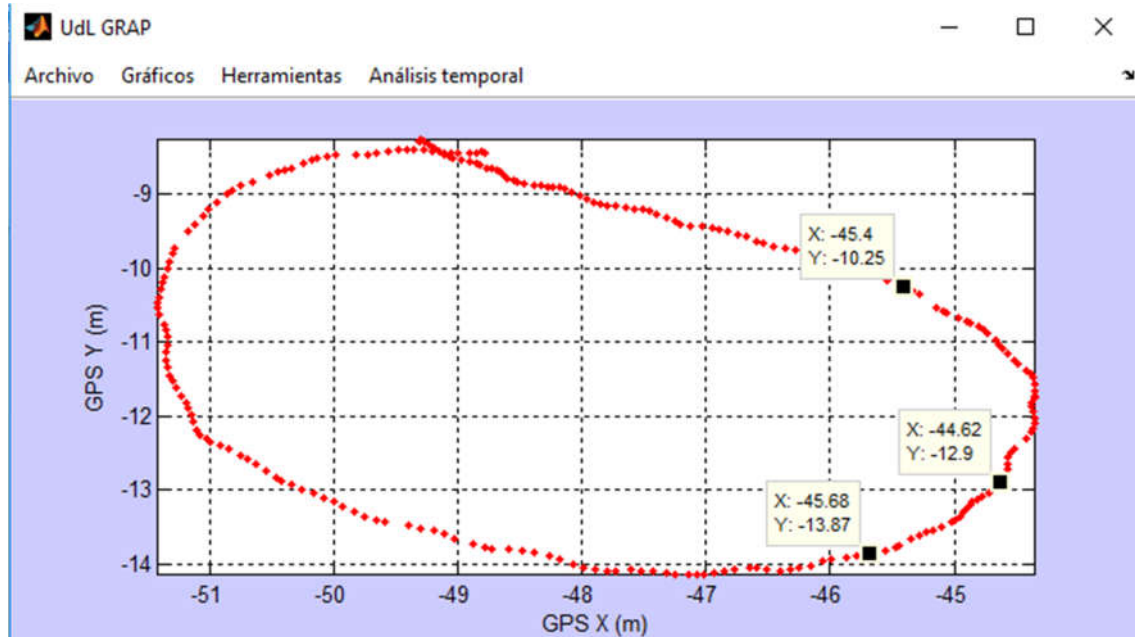


Figura 46: Visualización de datos múltiples con el cursor de datos

Por último, la cuarta pestaña es la de “Análisis temporal”, la cual, como su nombre indica, nos permitirá analizar los tiempos de recepción de datos y será una pieza fundamental para la mejora del código de adquisición.

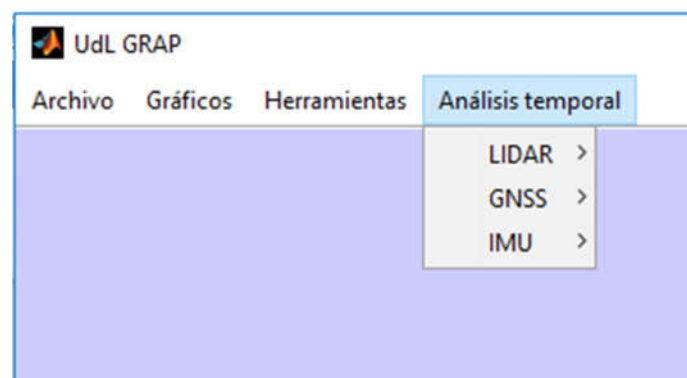


Figura 47: Menú Análisis temporal

Como se puede observar en la figura 47, el menú Análisis temporal cuenta con otros tres submenús:

- LiDAR:

Dentro de este submenú se podrán analizar los datos temporales del LiDAR, lo cual ayudará en gran medida a entender lo que está pasando en el proceso de recopilación de datos de este sensor.

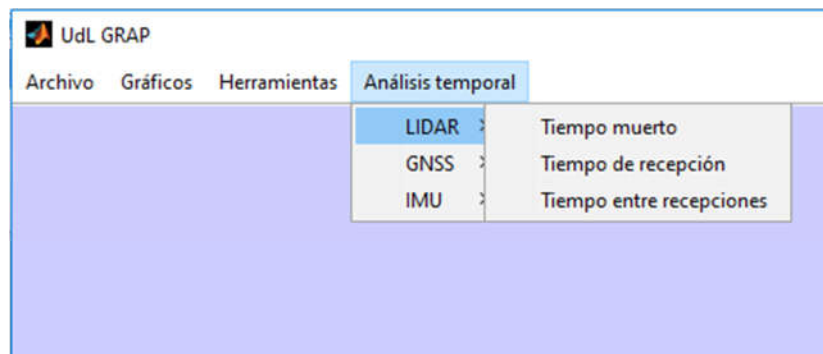


Figura 48: Menú Análisis temporal-LiDAR

Como se puede observar en la figura 48, el menú LiDAR cuenta con otros tres submenús:

○ Tiempo muerto:

Es el tiempo en el cual el microcontrolador no está realizando ninguna acción de preprocesamiento ni guardado con respecto al LiDAR. En otras palabras, es el tiempo entre que finaliza el guardado hasta que vuelve a recibir otro dato.

En la figura 49 se puede ver cómo se visualizaría la gráfica, teniendo en las ordenadas el tiempo transcurrido y en las abscisas el scan a analizar.

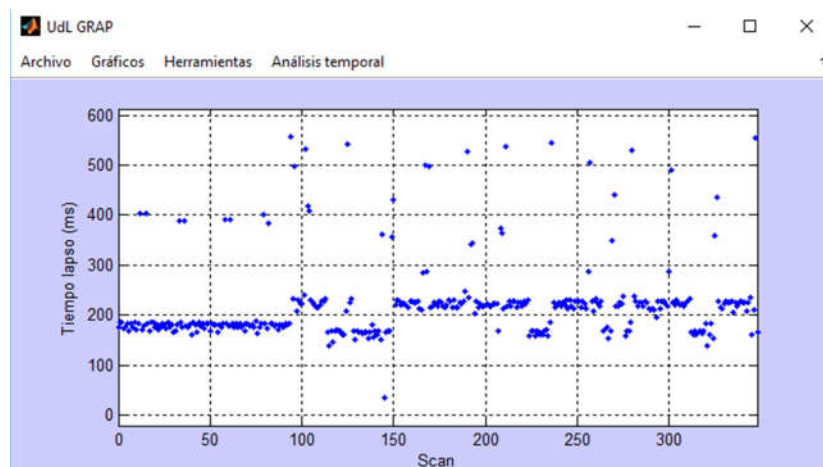


Figura 49: Graficado del tiempo muerto del sensor LiDAR

Este tiempo podría estar mostrando el margen temporal que se tiene para acelerar el proceso de petición de datos, aunque hay que tener en cuenta también el tiempo utilizado por los otros dos sensores.

- Tiempo de recepción:

Es el tiempo que tarda el microcontrolador en procesar y guardar todo el scan del LiDAR, es decir, desde la recepción del dato hasta la finalización del guardado de éste en la memoria SD.

En la figura 50 se puede ver cómo se visualizaría la gráfica, teniendo en las ordenadas el tiempo transcurrido y en las abscisas el scan a analizar.

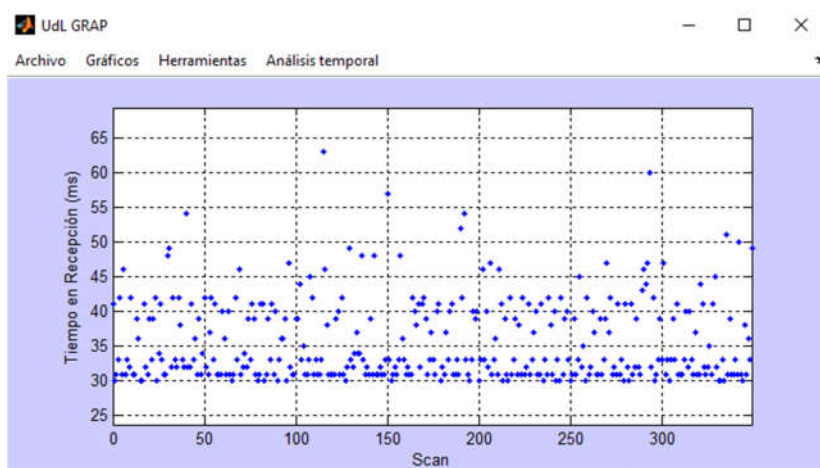


Figura 50: Graficado del tiempo de procesado y guardado del sensor LiDAR

La reducción de este tiempo es muy importante, sobre todo en las peticiones continuas con el comando ND, ya que cuanto más tiempo tarde esta parte del proceso, más datos se estarán dejando de adquirir y, por lo tanto, menos densidad tendrá la nube de puntos.

- Tiempo entre recepciones:

Es el tiempo entre el comienzo de una recepción y el de la siguiente, es decir, cada cuánto se recibe un nuevo dato LiDAR (o cada cuánto se es capaz de recibirlo).

En la figura 51 se puede ver cómo se visualizaría la gráfica, teniendo en las ordenadas el tiempo transcurrido y en las abscisas el scan a analizar.

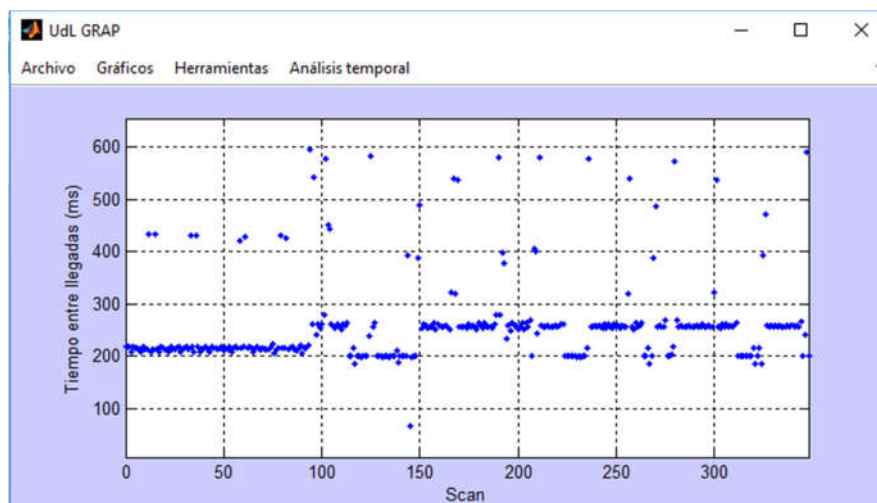


Figura 51: Tiempo entre recepciones del sensor LiDAR

La reducción de este tiempo es el objetivo final de las mejoras de software, ya que cuanto más pequeño sea este tiempo, como se ha dicho, mayor densidad de datos se obtendrán.

- GNSS:

Dentro de este submenú se podrán analizar los datos temporales del receptor GNSS, lo cual ayudará en gran medida a entender lo que está pasando en el proceso de recopilación de datos de este sensor.

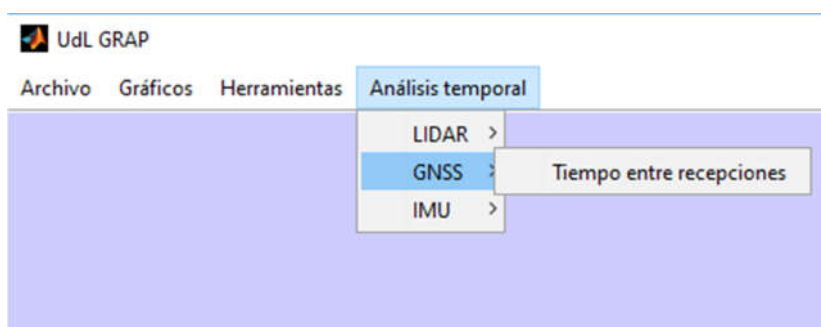


Figura 52: Menú Análisis temporal-GNSS

Para este sensor, al ser poca la cantidad de datos que procesa y guarda, tan solo se ve conveniente analizar el tiempo entre recepciones, como se muestra en la figura 52.

- Tiempo entre recepciones:

Es el tiempo entre el comienzo de una recepción y el de la siguiente, es decir, cada cuánto se recibe un nuevo dato del receptor GNSS-RTK (o cada cuánto se es capaz de recibirlo).

En la figura 53 se puede ver cómo se visualizaría la gráfica, teniendo en las ordenadas el tiempo transcurrido y en las abscisas el scan a analizar.

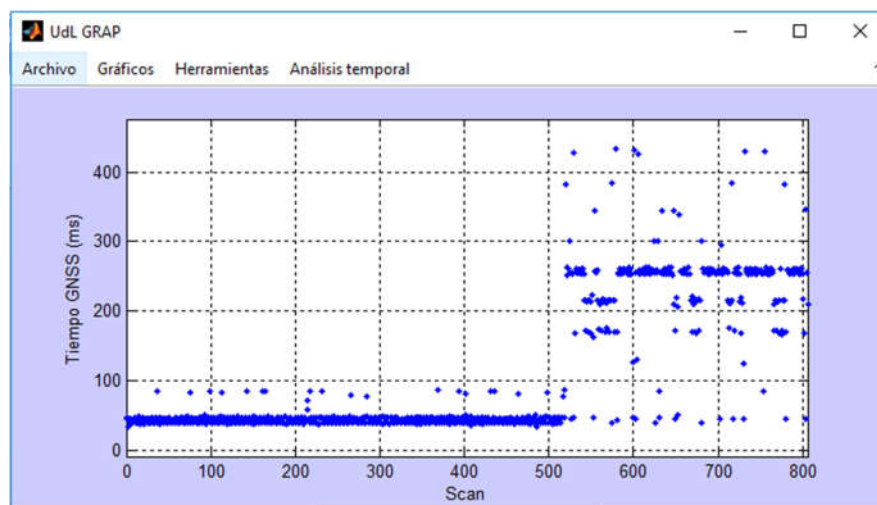


Figura 53: Tiempo entre recepciones del receptor GNSS-RTK

- IMU:



Figura 54: Menú Análisis temporal-IMU

Para la unidad de medida inercial, al igual que para el receptor GNSS-RTK, al ser poca la cantidad de datos que procesa y guarda, tan solo se ve conveniente analizar el tiempo entre recepciones, como se muestra en la figura 54.

- Tiempo entre recepciones:

Es el tiempo entre el comienzo de una recepción y el de la siguiente, es decir, cada cuánto se recibe un nuevo dato del receptor GNSS-RTK (o cada cuánto se es capaz de recibirlo).

En la figura 55 se puede ver cómo se visualizaría la gráfica, teniendo en las ordenadas el tiempo transcurrido y en las abscisas el scan a analizar.

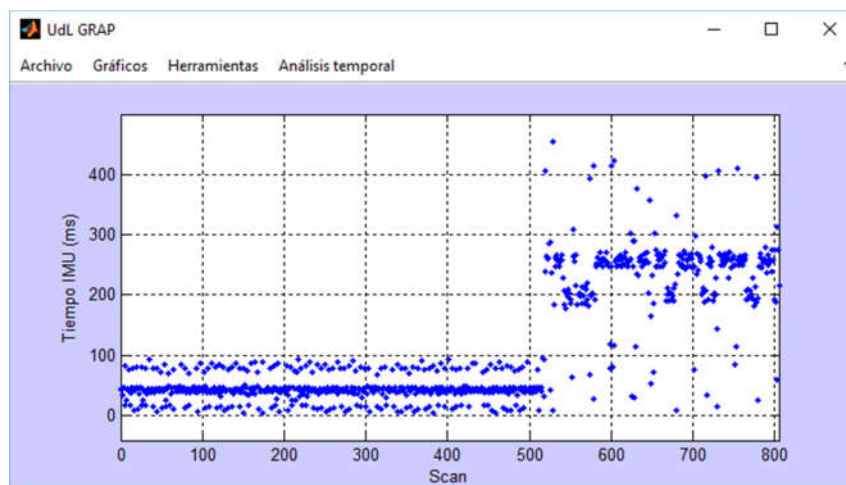


Figura 55: Tiempo entre recepciones de la unidad de medida inercial

Como ya se ha comentado, el menú, una vez todo está enlazado, no hay problema en modificar su aspecto. Es por eso que, modificando en algunos aspectos el programa y el menú, se han realizado programas paralelos para el análisis de cada sensor individualmente, ya que hay momentos en el que el análisis se centra en la mejora de adquisición de cada sensor por separado y, además, en un futuro puede interesar tener los sensores para otros objetivos y esto serviría para visualizar sus datos individualmente.

Por lo tanto, se tiene un código para el análisis de datos de los tres sensores, otro código para el análisis de datos del LiDAR individual, otro código para el análisis de los datos IMU individuales y por último otro para el análisis de los datos GNSS-RTK.

Los menús que tiene cada uno de estos códigos o scripts son reducidos al sensor, pero es válida la explicación de su funcionalidad realizada en las líneas anteriores.



2.3.3. Generación de la nube de puntos

El objetivo final del proyecto es la generación de una nube de puntos del entorno escaneado. Para ello, los datos de los tres sensores serán enviados como inputs a la función de reconstrucción, la cual realiza el siguiente proceso:

- Posicionamiento del sensor según dato del GNSS-RTK.
- Suma del ángulo marcado por la IMU y el ángulo correspondiente a la posición angular del dato del LiDAR.
- Aplicación de la distancia medida por el sensor LiDAR a la posición dada por el GNSS-RTK, utilizando la **matriz de Euler** para posicionar correctamente el nuevo punto en el espacio.
- Repetición de este proceso para todos los puntos escaneados.
- Guardado de estas coordenadas en la matriz final.

Una vez terminado este proceso para todos y cada uno de los puntos LiDAR recibidos, se crea un archivo de texto con las coordenadas XYZ de cada punto, de forma que CloudCompare, el siguiente software utilizado, entienda los datos que se le están enviando.

Paralelamente también se guardan los puntos del recorrido GNSS-RTK, para poder observar de forma más fácil si la reconstrucción es correcta, sobretodo ayudará a la corrección de los posibles errores entre el Norte Magnético y el Norte Geométrico.

Y como ya se ha adelantado, una vez guardado cada punto en el archivo de texto, se abre éste con el programa CloudCompare y allí se puede visualizar y realizar las comprobaciones pertinentes de los resultados obtenidos.

Ángulos de Euler:

Los ángulos de Euler constituyen un conjunto de tres coordenadas angulares, que sirven para especificar la orientación de un sistema de referencia de ejes ortogonales, normalmente móvil, respecto a otro sistema de referencia de ejes ortogonales normalmente fijos.

Tres ángulos, que mediante una sucesión ordenada de giros, definen el cambio de un sistema de coordenadas a otro. Los ángulos de Euler (phi, theta, psi) corresponden con los ángulos convencionales de 'roll' (Φ), 'pitch' (Θ), 'yaw' (Ψ) que se utilizan en navegación para especificar la actitud de un móvil. Poniendo el ejemplo de un avión, como se puede observar en la figura 56, el ángulo de 'roll' (Φ) es el giro de las alas del avión, el ángulo de 'pitch' (Θ) la inclinación del morro y el ángulo de 'yaw' (Ψ) es el giro del morro del avión respecto al Norte.

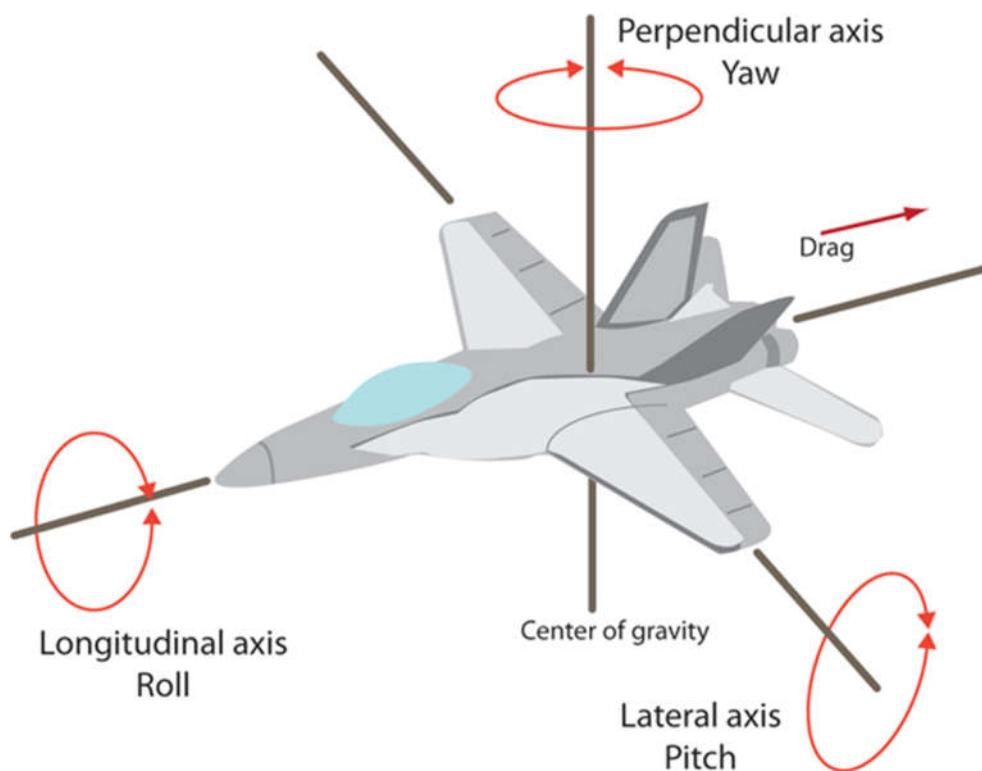


Figura 56: Explicación gráfica de los ángulos roll, pitch y yaw

Es importante señalar que los ángulos de Euler no están únicamente definidos y existen ambigüedades. Por eso se deben hacer las rotaciones siempre en el mismo orden ya que los mismos ángulos, si son aplicados con órdenes diferentes, dan lugar a transformaciones diferentes.

Matriz de rotación:

La matriz de rotación (también conocida como 'Direction Cosine Matrix', DCM) es una representación muy conocida, redundante y completa de la orientación. La matriz de rotación se puede interpretar como los componentes del vector unitario del sistema de coordenadas del sensor (**S**) expresado en **G** (véase figura 57). Para el vector unitario R_{GS} de **S** se encuentra en las columnas de la matriz, por lo que la columna 1 es X_s expresado en **G**, etc. La normal de la matriz de rotación es siempre igual a uno (1) y una rotación de R_{GS} seguido por el inverso de rotación R_{SG} , se obtiene la matriz identidad I^3 .

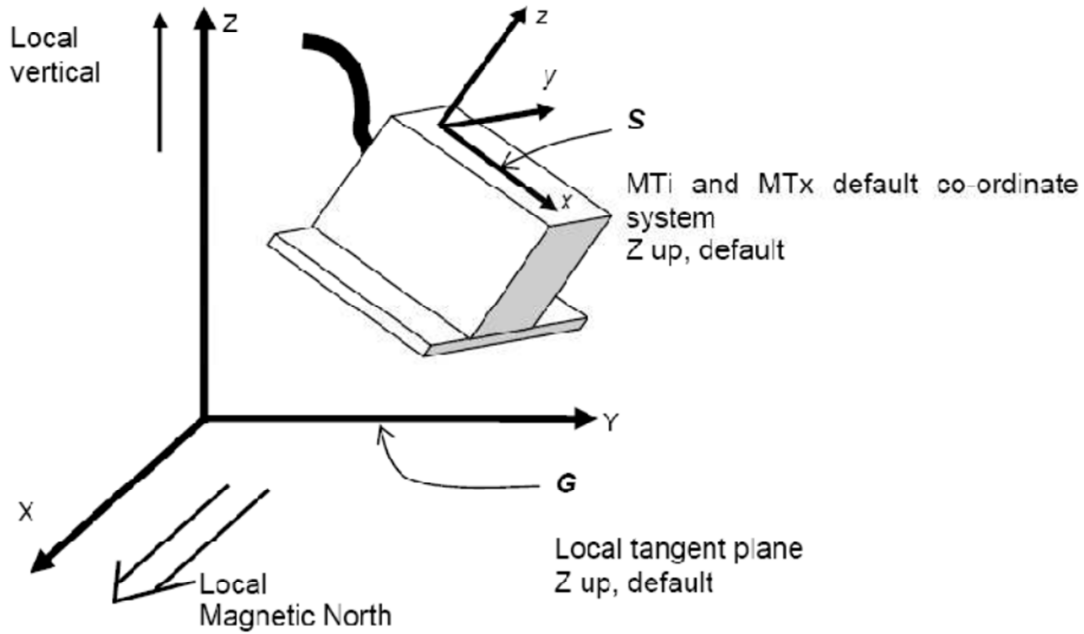


Figura 57: MTi en el sistema de coordenadas de la Tierra(Fuente:Xsens)

La figura 58 muestra la expresión de la matriz de rotación cuando los datos recibidos son los ángulos de Euler:

$$R_{GS} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}$$

Figura 58: Matriz de rotación para ángulos de Euler



Transformación en Matlab:

Para realizar la transformación de los datos recibidos en una nube de puntos, el script Matlab utiliza la función de transformación “*EulerAngles*”, cuya diagrama de flujo se muestra en la figura 59.



Figura 59: Diagrama de flujo de la función *EulerAngles*

Este bucle corre cada vez que recorriendo la matriz de datos guardada en la tarjeta SD el script encuentra un marcador de dato LiDAR ($L = 18200$). En ese momento, tal y como se han guardado los datos, ya tiene almacenado un dato del sensor inercial y otro del receptor GNSS-RTK, con lo que puede proceder a realizar las operaciones que se muestran en la figura 59.

Este bucle va guardando cada una de las coordenadas corregidas en una variable hasta que se ha recorrido todo el fichero de datos, y es entonces cuando se puede realizar la visualización de éstos o bien elegir la opción de guardar y así tenerlos disponibles en un archivo de texto.



El guardado en un archivo de texto, entre otras cosas, permite realizar la visualización a través del programa CloudCompare y así poder realizar comprobaciones de la reconstrucción realizada.

2.3.4. Validación de la nube de puntos

Para la validación de la nube de puntos se realizarán dos pruebas.

Validación 1

La primera prueba será la comparación de las dimensiones a partir de la medición de figuras geométricas conocidas. Como se muestra en la figura 60, estas figuras son un prisma rectangular y un cilindro, de los cuales se pueden tomar fácilmente medidas y compararlas posteriormente con las medidas que se tomen con el CloudCompare, y así tomar una decisión de la precisión del sistema completo (la mezcla entre la adquisición de datos y el tratamiento de éstos).



Figura 60: Figuras geométricas desde diferentes ángulos preparadas para la primera prueba de validación

Las dimensiones de las figuras geométricas utilizadas en esta primera prueba son:

Dimensiones del cilindro:

Altura = 0.95 m

Diámetro = 0.49 m

Dimensiones del prisma rectangular:

Altura: 1,00 m

Lado 1: 0.50 m

Lado2: 0.50 m

Validación 2

La segunda prueba será la comparación de distancias entre ejes, alturas y longitudes de los diferentes elementos de un “Opel Astra 5p (2004)” mostrado en la figura 61, del que se dispone de la ficha técnica.



Figura 61: Opel Astra 5p (2004)

Las dimensiones del vehículo son:

- Longitud: 4,25 m
- Altura: 1,46 m
- Anchura: 1,75 m
- Batalla¹: 2,61 m

Las comprobaciones dimensionales son analizadas en el apartado 3.2 *Nubes de puntos*.

¹ Distancia entre eje delantero y trasero del vehículo



3. Resultados y discusión

3.1. Tiempos de adquisición y sincronización

El objetivo principal de este apartado es el de conseguir una adquisición de datos rápida y sincronizada de los tres sensores a la vez, pero como mayor hándicap se tiene el bloqueo del microcontrolador cuando se están adquiriendo estos datos.

Para poder avanzar en este bloque, se analiza a fondo el comportamiento temporal tanto de los sensores por separado como de los sensores trabajando conjuntamente, y para ello se utiliza el programa de apoyo al análisis descrito con anterioridad en el apartado 2.3.2.

ADQUISICIÓN LiDAR

Se debe tomar la decisión de escoger un tipo de comunicación continua (comando ND) o una comunicación con petición previa en cada dato (comando HD). Para eso, se realiza un estudio del tiempo de adquisición en cada caso, el cual marque de forma empírica la elección adecuada.

Envío continuo de datos (Comando ND)

El estudio realizado consiste en visualizar los diferentes tiempos generados por el programa de adquisición de datos del LiDAR cuando se selecciona la opción de envío de datos continuo.

En la opción de envío continuo de datos, existe la forma de configurar el LiDAR para que envíe todos los scans que realiza o bien que salte cierto número de scans entre envío y envío (skips). Esta herramienta puede servir para dar más tiempo al programa de cara al procesado de los datos recibidos.

ND Skips 0:

En primer lugar, se analizan los tiempos si se configura de forma que el LiDAR envíe todos los scans que hace.

En la figura 62, se puede observar el tiempo entre la recepción de un scan y el siguiente.

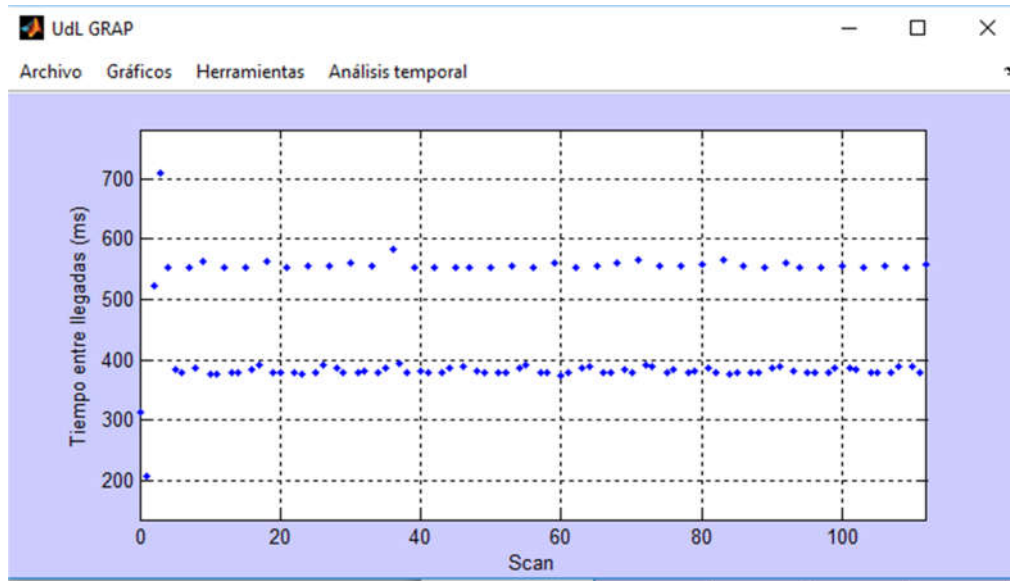


Figura 62: Tiempo entre recepciones del sensor LiDAR [ND - sin skips]

Según el fabricante, el LiDAR hace un scan cada 25 milisegundos, con lo que se puede ver con claridad que el software, a una velocidad de envío tan rápida por parte del LiDAR, no es capaz de recibir los datos.

No se tiene clara la razón de que aun así los tiempos de recepción sean tan “estables”, pero de todas formas la adquisición se bloquea en menos de medio minuto.

Por otra parte, con esta misma configuración, se debe estudiar los tiempos de guardado, los cuales se muestran en la figura 63.

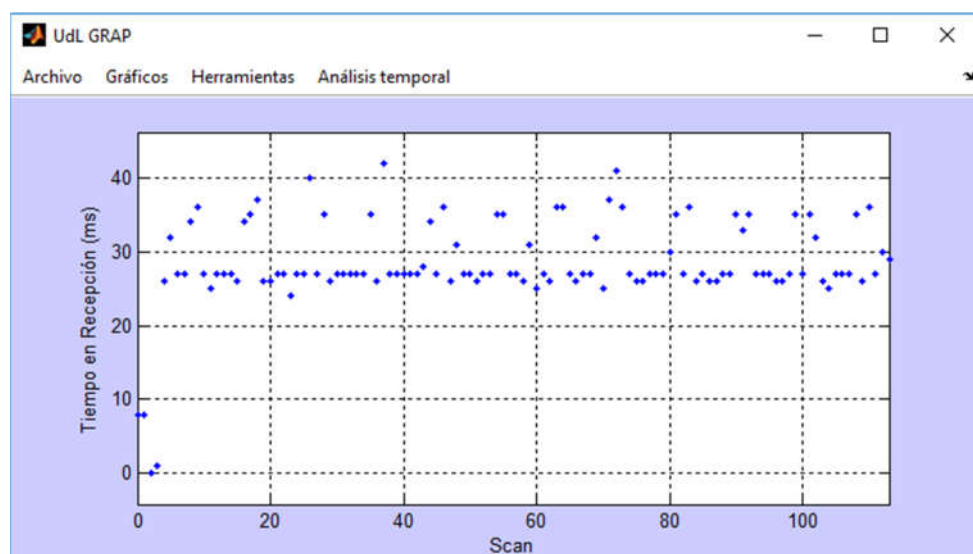


Figura 63: Tiempo de procesado y guardado de sensor LiDAR [ND - sin skips]

Estos tiempos son mayores que los 25 milisegundos con los que el LiDAR envía datos, con lo que podría ser el motivo del error analizado en la figura 62.

ND Skips 3:

Para evitar que los datos lleguen tan rápido, se hace la prueba dándole tres skips al comando ND para que, entre un dato y el siguiente que envíe el LiDAR, deje tres scans sin enviar.

En la figura 64 se puede ver el tiempo entre recepciones de datos.

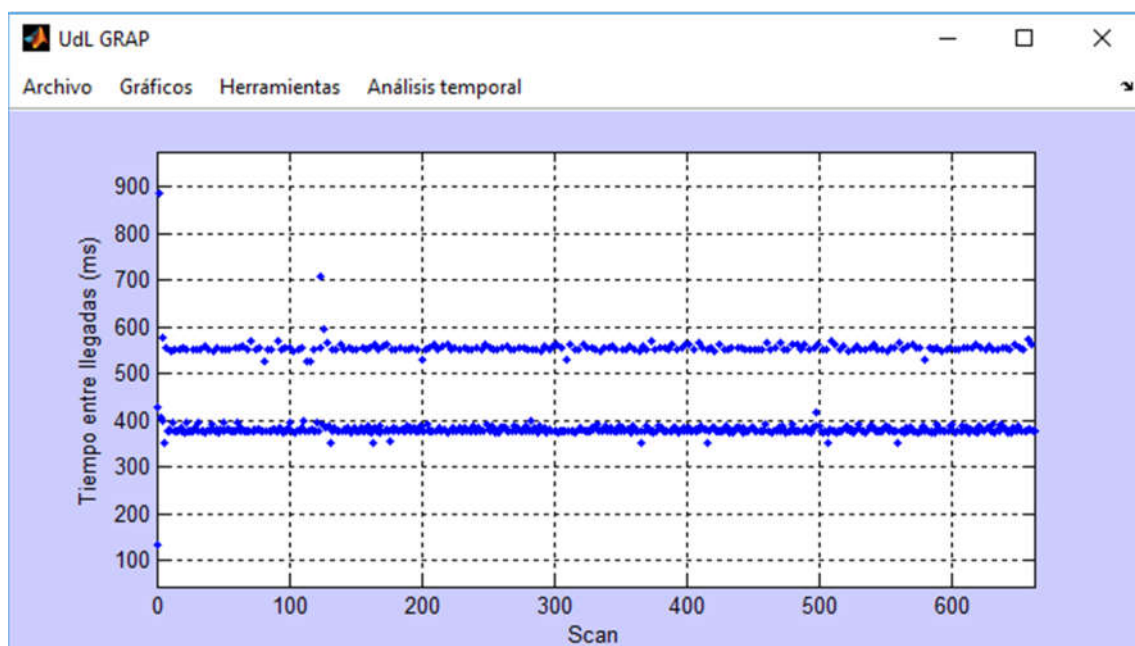


Figura 64: Tiempo entre recepciones del sensor LiDAR [ND – 3 skips]

Vuelven a salir los mismos tiempos que en el caso anterior, cuando se recibían todos los datos, la única diferencia es que en este caso el programa no se bloquea.

Se puede observar que los tiempos, aunque altos, siguen siendo estables en las mismas franjas.

Se analiza el tiempo de procesado y guardado de cada muestra a través de la figura 65.

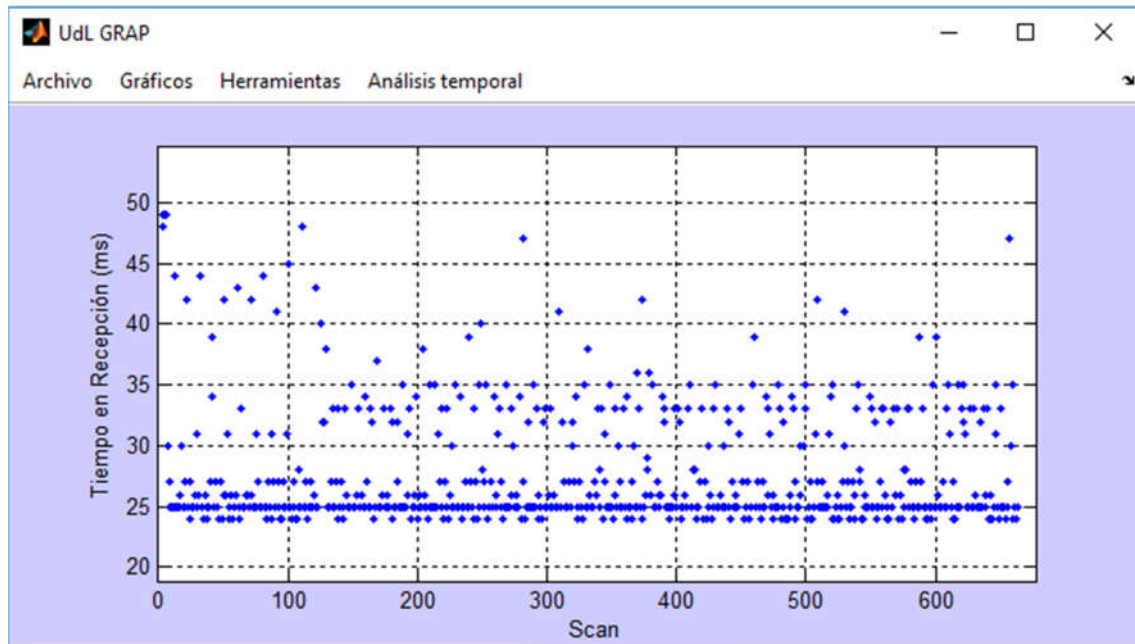


Figura 65: Tiempo de procesamiento y guardado de sensor LiDAR [ND – 3 skips]

En el caso anterior, el análisis de este error era que el guardado era más costoso en tiempo que lo que le costaba al LiDAR volver a enviar otro dato. En este caso, habiendo tres saltos, no es posible que este sea el motivo del “error”, ya que en todos los tiempos de guardado están por debajo de los 50 milisegundos, y el LiDAR envía los datos, teóricamente, cada 100 milisegundos.

Se desconoce por completo el motivo de este efecto, se ha estudiado con detenimiento pero no se ha conseguido llegar ni a la explicación ni a la solución del mismo.

ND Skip 9:

Para hacer más visible el alcance del error que se visualiza en los análisis anteriores, se muestra el resultado con 9 saltos entre muestra y muestra.

En la figura 66 se puede ver el resultado del tiempo de recepción de datos.

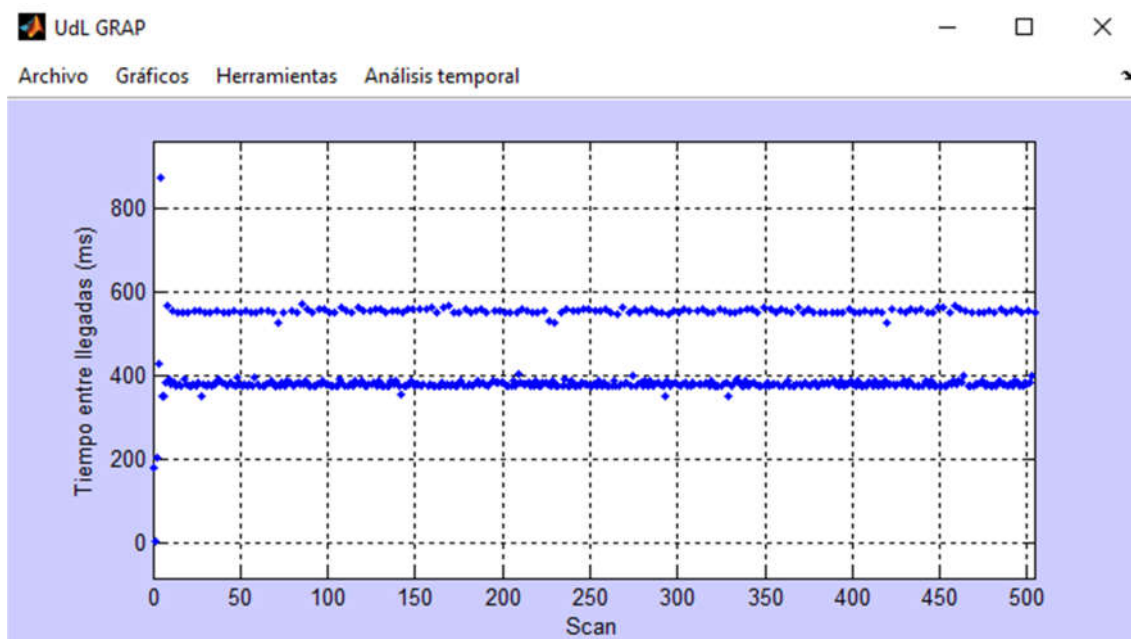


Figura 66: Tiempo entre recepciones del sensor LiDAR [ND – 9 skips]

Como se observa de nuevo, los tiempos de recepción entre datos vuelven a marcar la misma pauta, cuando teóricamente el LiDAR está enviando los datos cada más de 200 milisegundos.

En la figura 67 se puede analizar el tiempo de procesado y guardado.

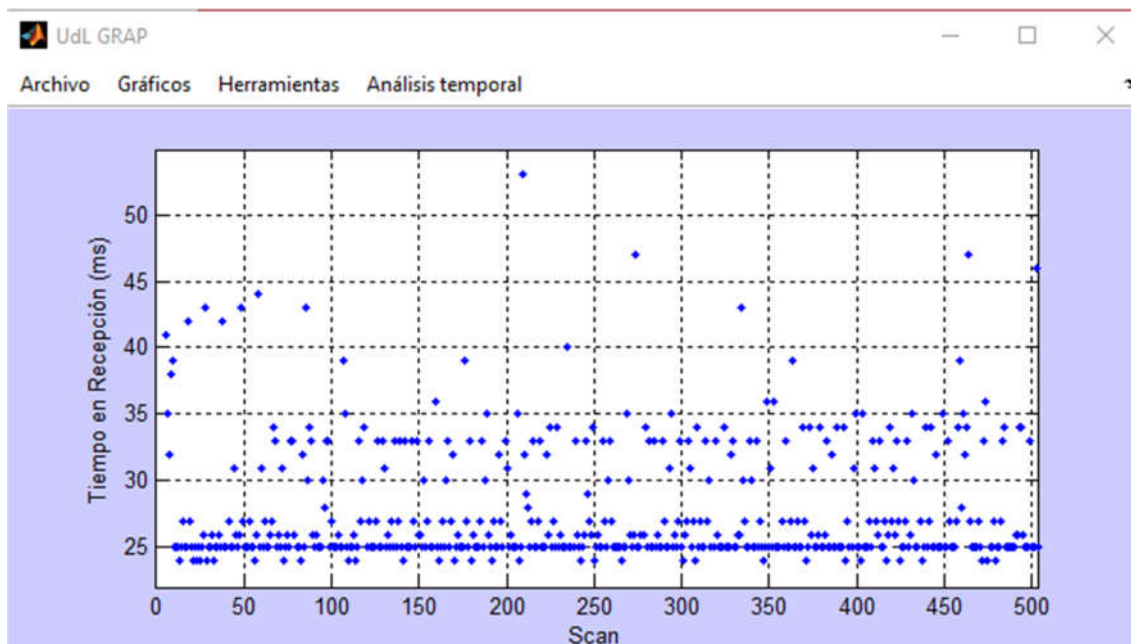


Figura 67: Tiempo de procesamiento y guardado de sensor LiDAR [ND - 9 skips]

Como en el resto de casos, los tiempo de guardado están por debajo de los 50 milisegundos, y se concentran más en la franja entre los 25 y 35 milisegundos.

Se ha probado con todos los skips, desde cero hasta nueve, y el resultado es el mismo en todos.

Para asegurar que la orden de hacer skips es entendido por el LiDAR, se realiza una prueba sin procesado ni guardado de datos, tan solo, cuando se recibe un dato LiDAR, se guarda el tiempo de recepción de éste.

ND skip 1 (sin guardado):

Se prueba, en primer lugar, dando tan solo un skip. En la figura 68 se puede analizar el efecto de esta instrucción.

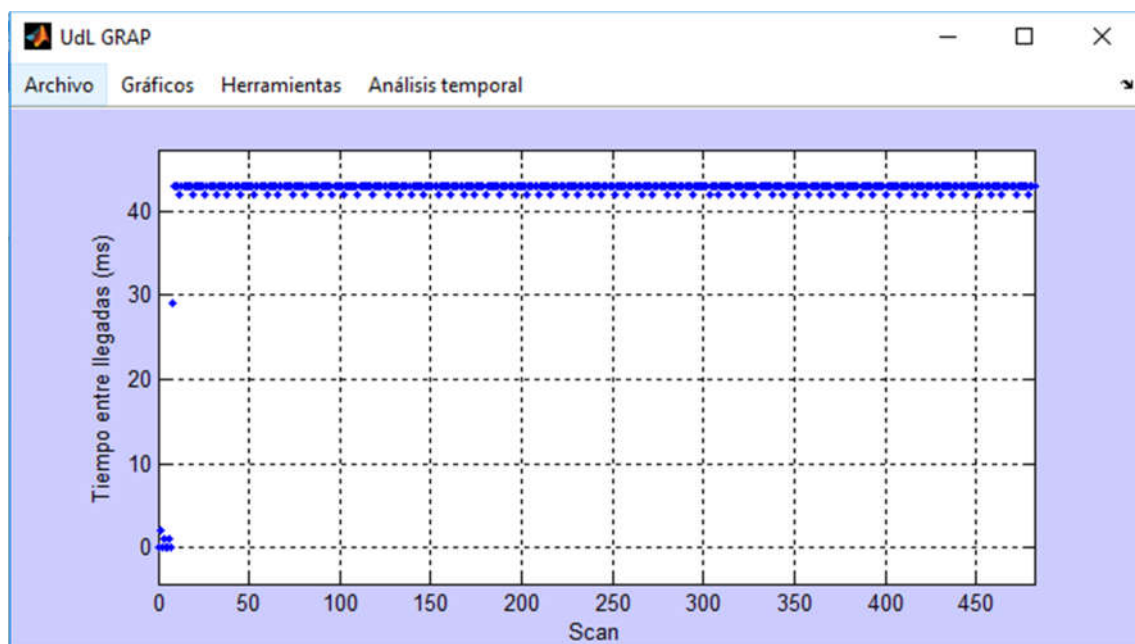


Figura 68: Tiempo entre recepciones del sensor LiDAR [ND – 1 skip (sin guardado)]

Efectivamente, como se puede comprobar, el tiempo entre recepciones se ha doblado. No es exactamente 50 milisegundos porque, al parecer, la frecuencia de envío es algo más elevada que los 40 Hz que especifica el fabricante.

En la figura 69 se puede comprobar que el tiempo de guardado es prácticamente nulo excepto en algún caso excepcional, seguramente debido a la comunicación con la memoria SD.

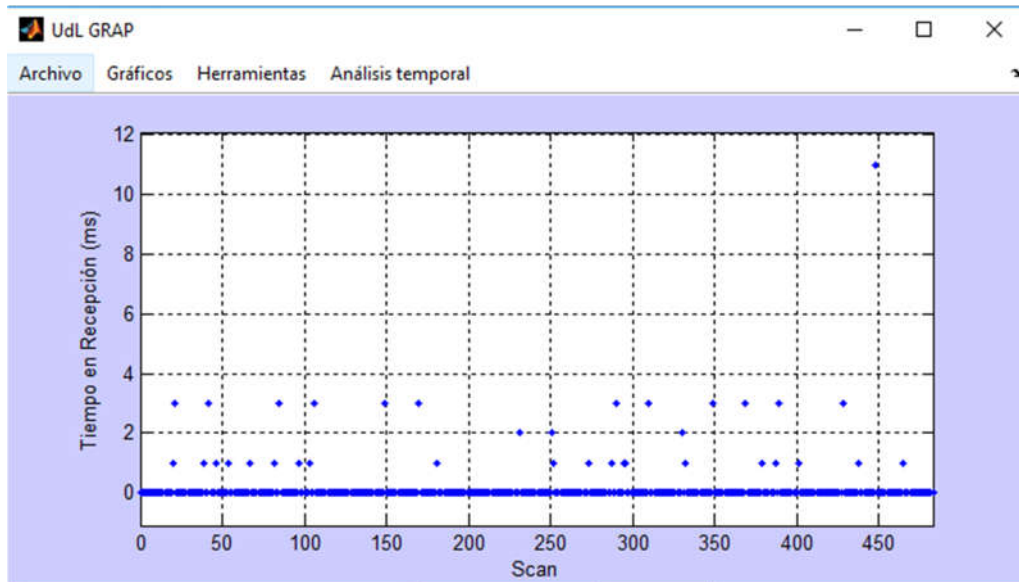


Figura 69: Tiempo de guardado de timer en SD [ND – 1 skip]

ND skip 3 (sin guardado):

Como se ha visto que el tiempo de envío es menor que el estipulado por el fabricante, se comprueba que dando tres skips el tiempo de llegada de datos es mayor que el tiempo de guardado visto en la figura 67.

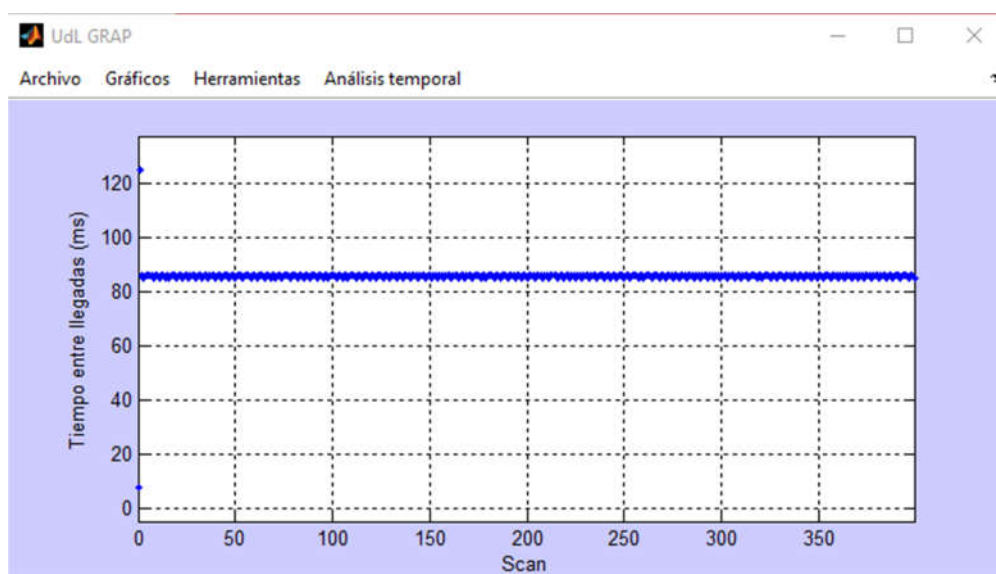


Figura 70: Tiempo entre recepciones del sensor LiDAR [ND – 3 skips (sin guardado)]

Como se puede en la figura 70, efectivamente el tiempo entre recepciones es menor que los 100 milisegundos teóricos, pero aun así es un tiempo lo suficiente amplio como para que el microcontrolador realice el procesado y guardado con holgura.

En la figura 71 se puede comprobar que el tiempo de guardado es prácticamente nulo excepto en algún caso excepcional, seguramente debido a la comunicación con la memoria SD.

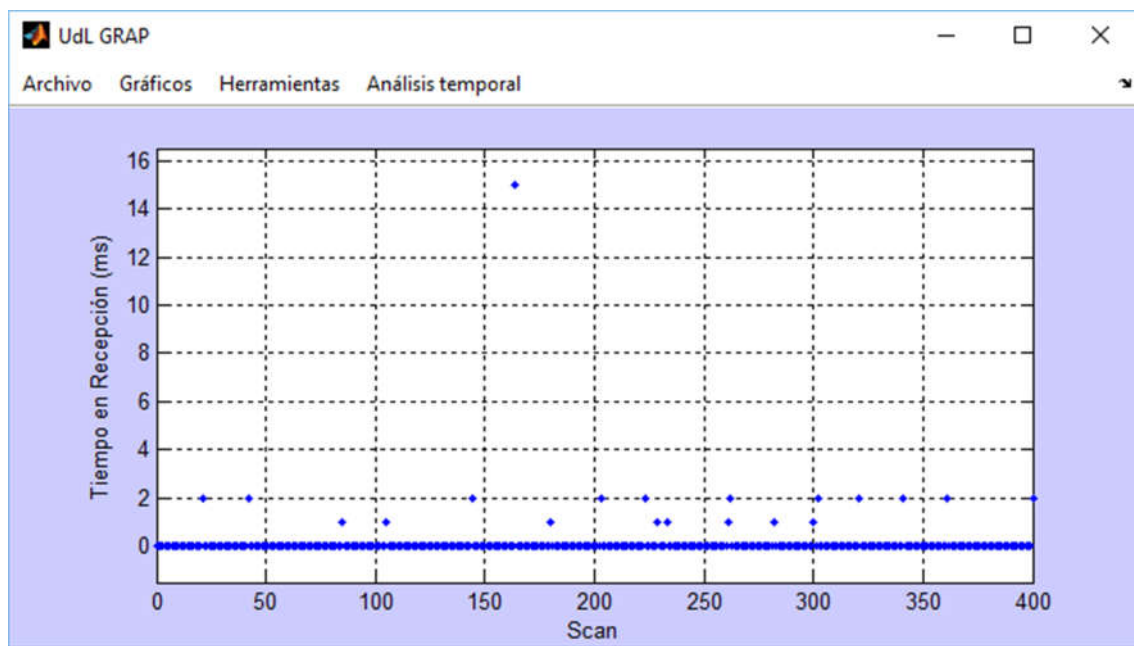


Figura 71: Tiempo de guardado de timer en SD [ND – 3 skips]

Envío de datos tras petición (orden HD):

Se han realizado diferentes pruebas, se vio que con una espera menor de 100 ms el programa se bloqueaba, con lo que no se pueden mostrar gráficos ya que al bloquearse no quedan guardados los datos.

Poniendo una espera de 200 milisegundos entre dato y dato, y en caso de no recibir el dato en 100 milisegundos volver a pedir uno, se tienen los siguientes resultados:

En esta ocasión se realiza un análisis uniando varias figuras, ya que se considera que el análisis entrelazado dejará una visión más clara de lo que sucede en el proceso.

En la figura 73, ampliación de la figura 72, se puede analizar el tiempo entre recepciones, lo cual, unido a la figura 75, ampliación de la figura 74, que muestra el tiempo de procesado y guardado de las muestras, y la figura 76, que muestra el tiempo en el que el procesador está a la espera de dato (tiempo muerto), deja un resultado claro.

El “tiempo muerto” mínimo es 200 milisegundos, exactamente el tiempo que se le ha dado de espera para la petición de un nuevo dato; si, a ese tiempo, se le suma el tiempo de procesado y guardado, donde la franja de tiempos está entre los 30 y 40 milisegundos, se obtiene el tiempo adquirido entre recepciones.

Una posible conclusión de esto sabiendo, además, que con esta configuración la adquisición no se bloquea, es que el dispositivo de adquisición está trabajando de forma estable y sin problemas.

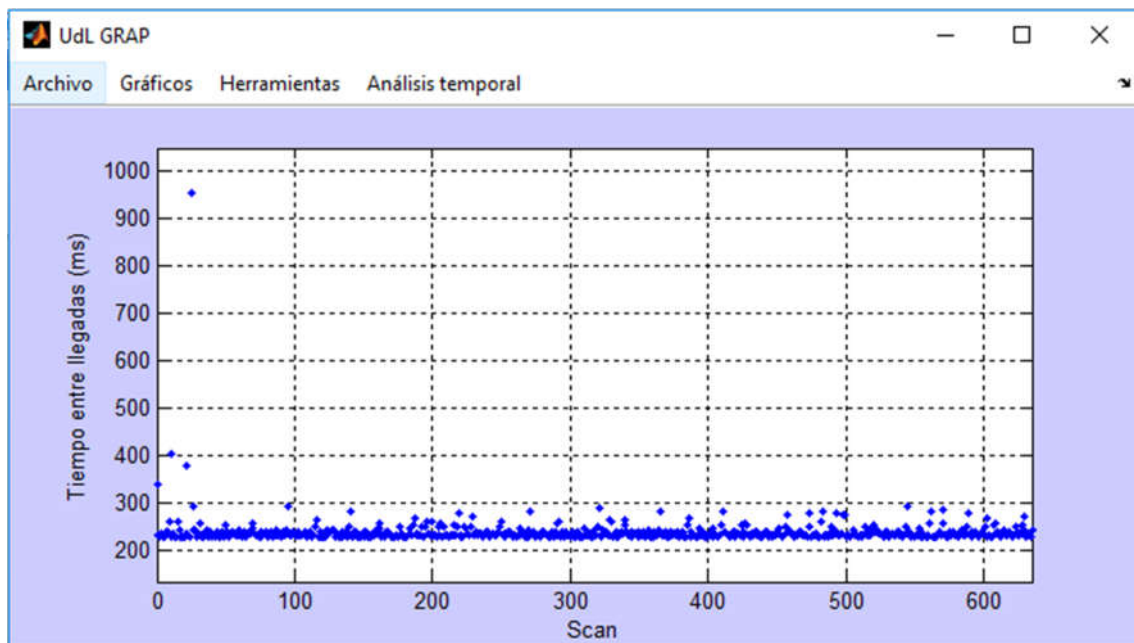


Figura 72: Tiempo entre recepciones de sensor LiDAR [HD]

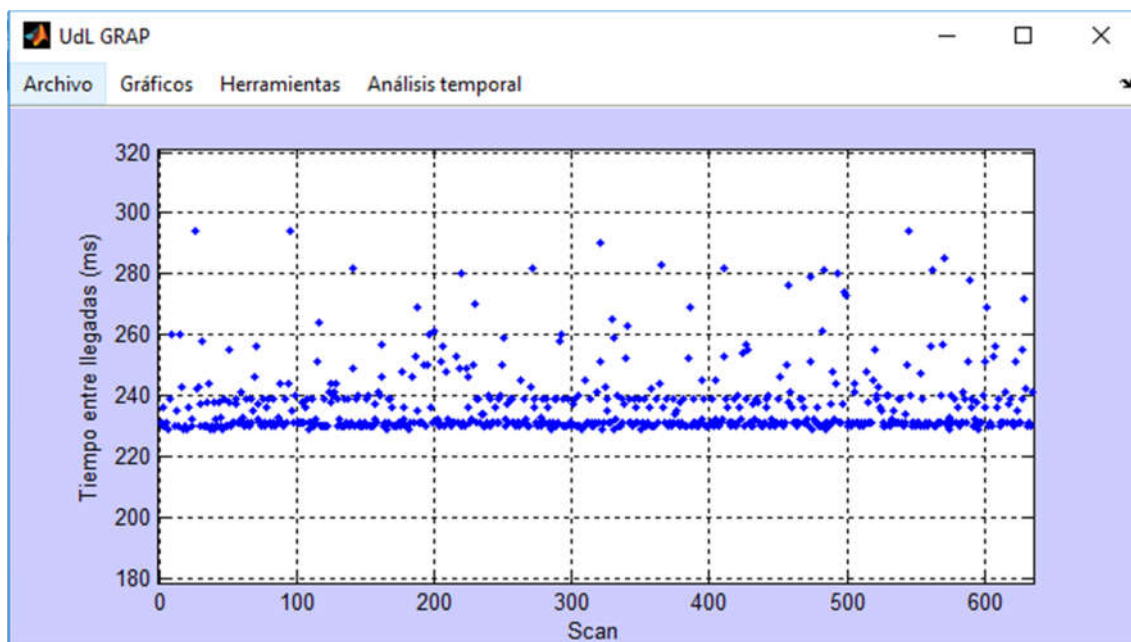


Figura 73: Ampliación del eje Y de la gráfica de la figura 72

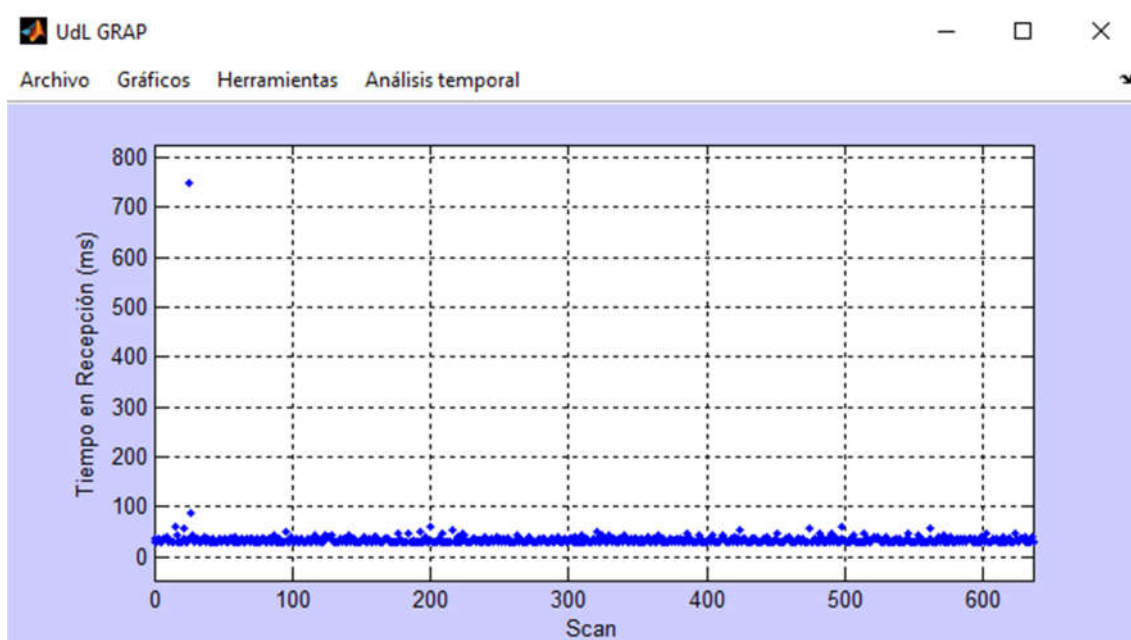


Figura 74: Tiempo de procesado y guardado de sensor LiDAR [HD]

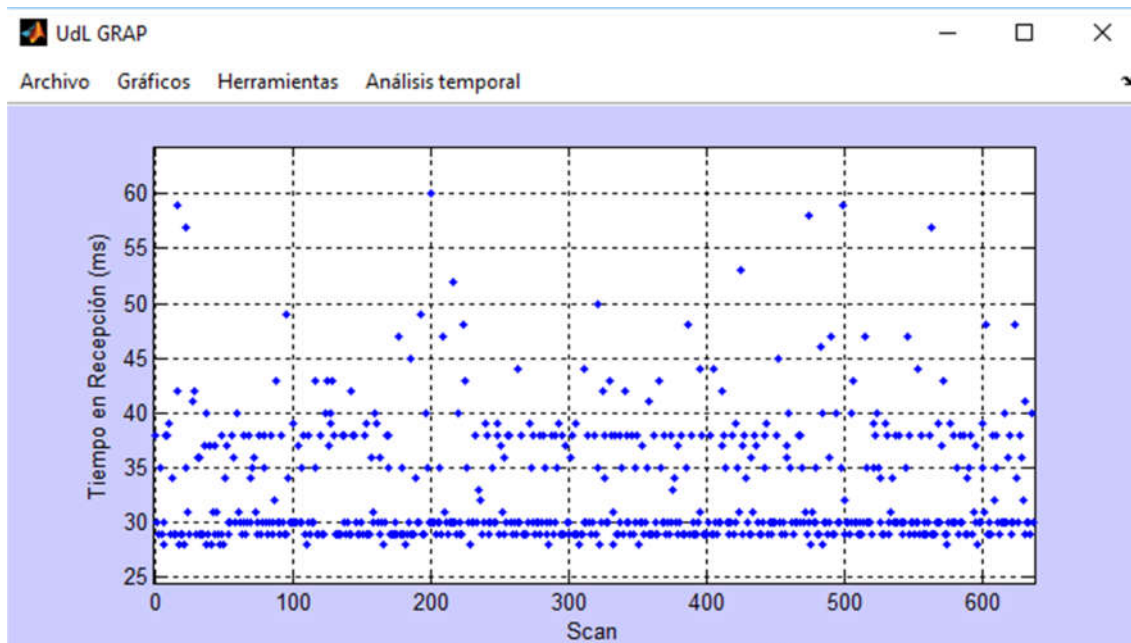


Figura 75: Ampliación del eje Y de la gráfica de la figura 74

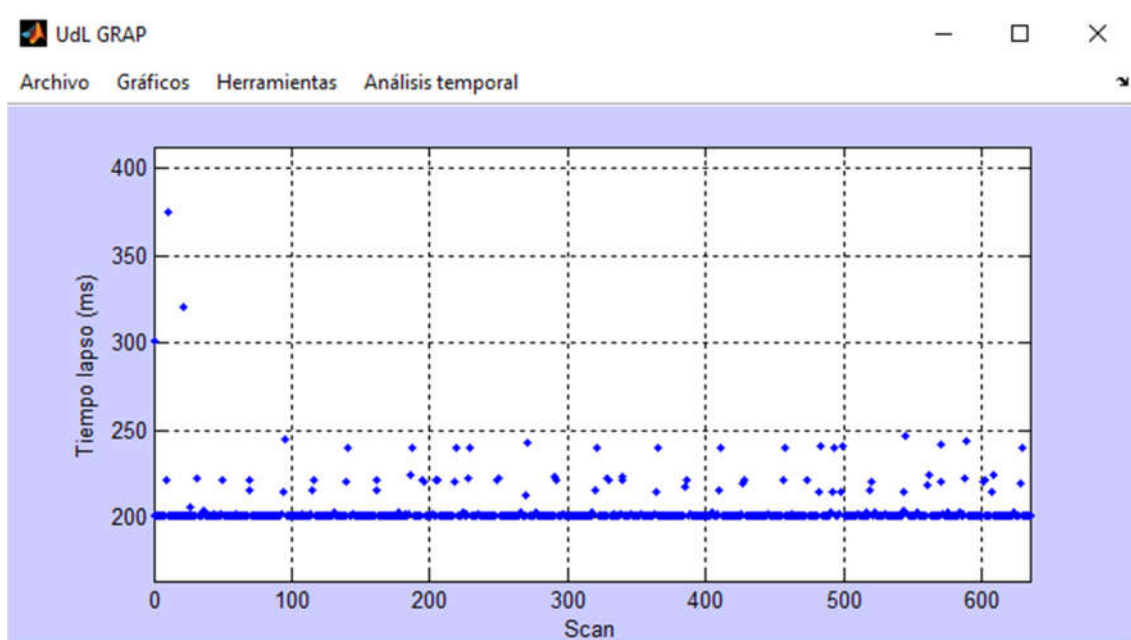


Figura 76: Tiempo muerte del sensor LiDAR [HD]

Guardado de datos sin proceso previo:

Se realiza un estudio del tiempo que se ahorraría si el guardado de los datos recibidos del sensor LiDAR fuese directo y no con un procesado previo.

Los resultados no se pueden graficar, ya que las dimensiones del guardado son complejas y no se adaptan al código de RStudio que realiza la preparación de los datos para abrirlos posteriormente con Matlab, pero se han analizado uno a uno leyéndolos a través del Wordpad.

Los tiempos entre recepciones son de entre 5 y 9 milisegundos, lo cual habla de un ahorro de más del 70% del tiempo.

Esto ayudaría en gran medida a la mejora del código, ya que podría ser la solución al problema con el comando ND o bien un gran ahorro de tiempo que permitiría realizar peticiones más a menudo con el comando HD.

ADQUISICIÓN SIMULTÁNEA SENSOR INERCIAL Y RECEPTOR GNSS-RTK

Se realiza la adquisición simultánea de estos dos sensores ya que, siendo mínimo su tiempo de procesado, nos da una idea muy aproximada de cómo se comportan individualmente.

Unidad de medida inercial:

En la figura 77 se muestra el tiempo entre recepciones de la unidad de medida inercial.

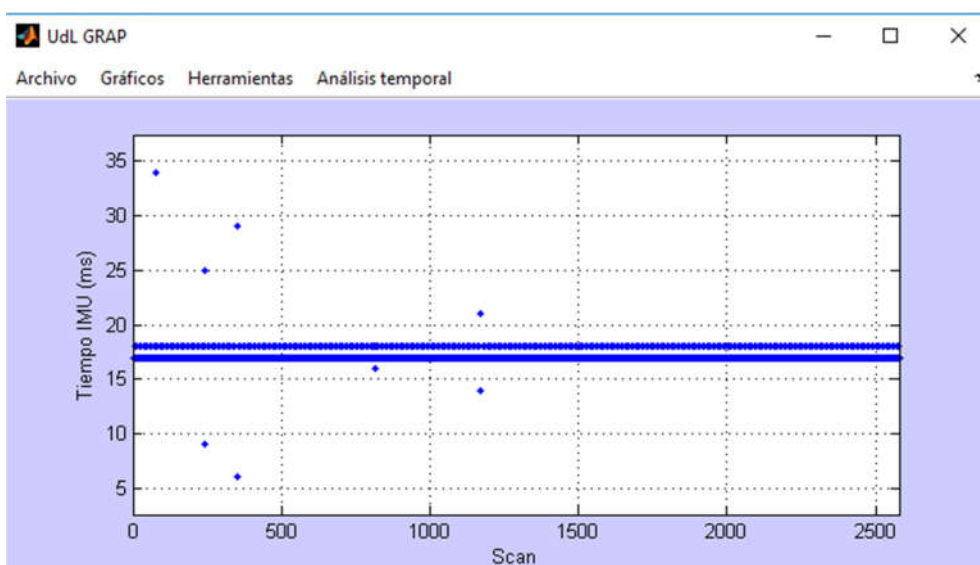


Figura 77: Tiempo entre recepciones de la unidad de medida inercial

Como se puede apreciar mejor en la figura 78, ampliación del eje Y de la gráfica de la figura 77, el tiempo entre recepciones de este sensor es de entre 17 y 18 milisegundos.

Se debe hacer la observación de que el Timer 6, que es el utilizado para la adquisición de los diferentes tiempos en el proceso de adquisición, está configurado para que de precisión de milisegundos, y es por eso que se ven esas dos bandas tan marcadas.

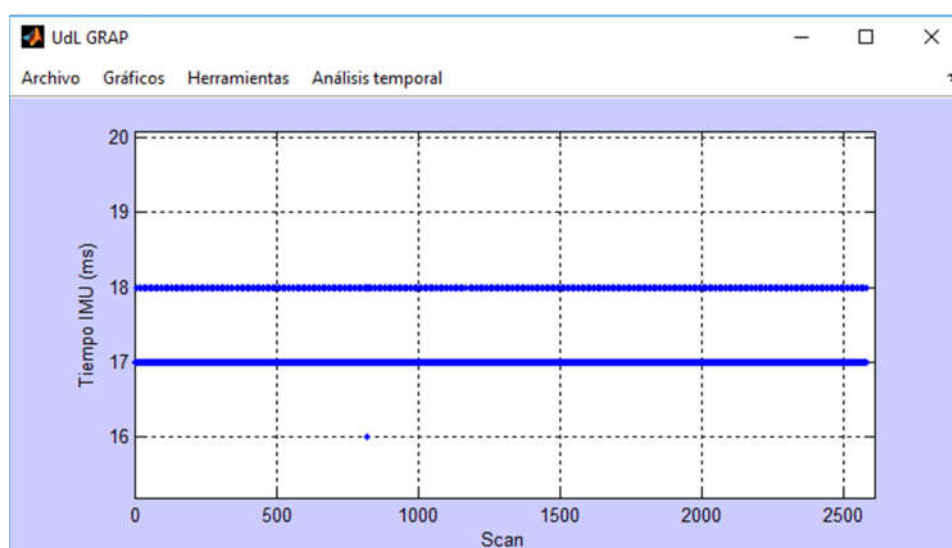


Figura 78: Ampliación del eje Y de la gráfica de la figura 77

Receptor GNSS-RTK:

En la figura 79 se muestra el tiempo entre recepciones del receptor GNSS-RTK.

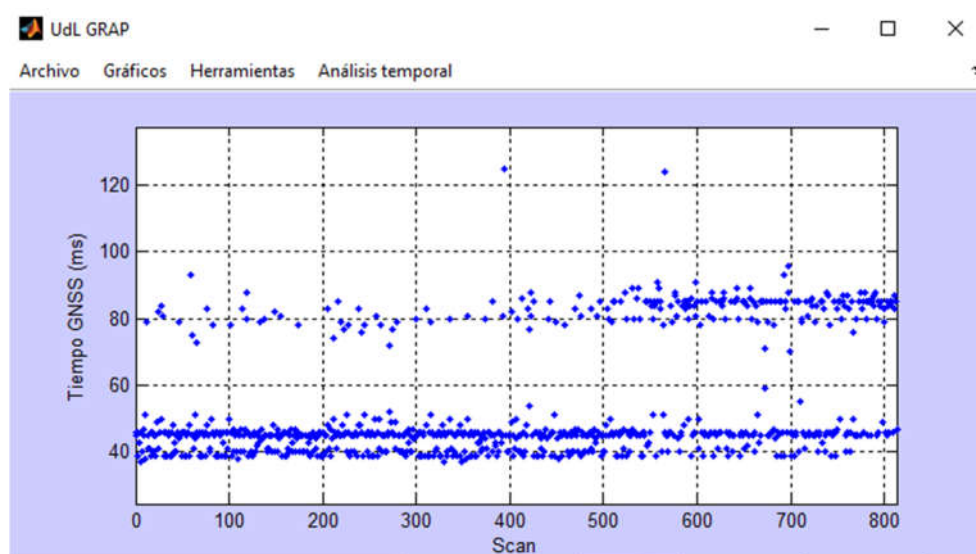


Figura 79: Tiempo entre recepciones del receptor GNSS-RTK

Se puede observar que la franja de tiempo predominante es entre 40 y 50 milisegundos, tal como especifica el fabricante, ya que ofrece un valor de 20Hz (recepciones cada 50 milisegundos).

Por otro lado, se ve otra franja importante alrededor de los 85 milisegundos, la cual se aprecia con más claridad en la figura 80, ampliación de la anterior. Esto puede ser debido a pérdida de datos por error de transmisión por causas que se desconocen por el momento.

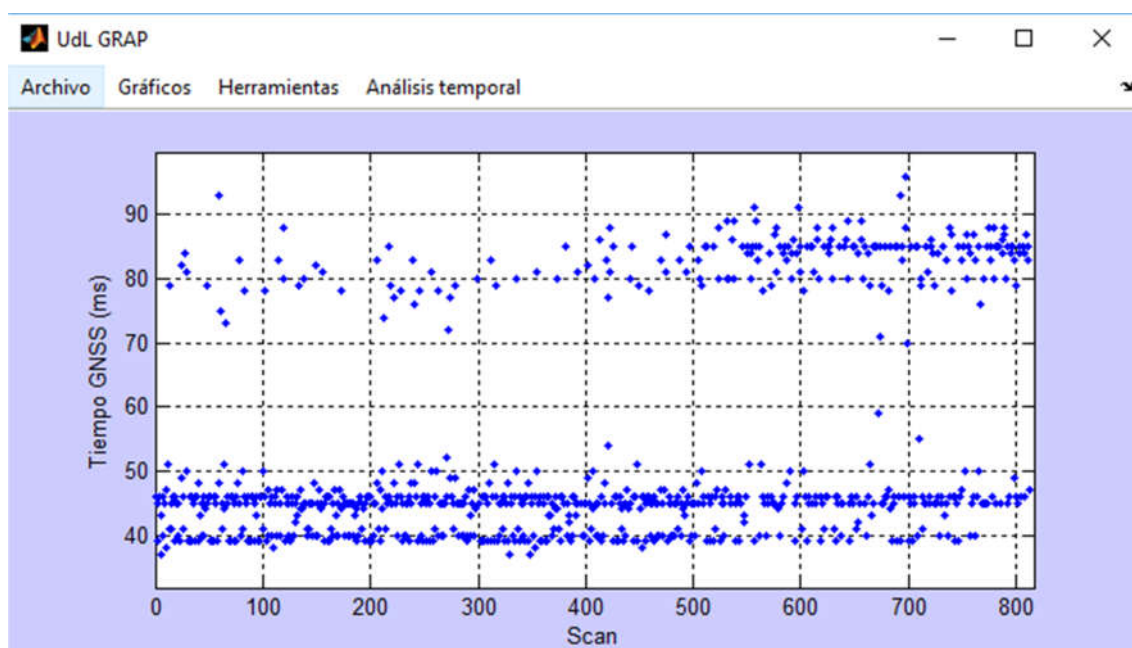


Figura 80: Ampliación del eje Y de la gráfica de la figura 79

ADQUISICIÓN SIMULTÁNEA DE LOS TRES SENSORES

Se analiza las dos adquisiciones que mejor referencian los objetivos a conseguir dentro de este apartado, una es conseguir un software estable y que no se bloquee (a partir de ahora “Objetivo 1”) y la otra conseguir la mayor cantidad de datos posibles en la adquisición, para así, por lo tanto, conseguir una densidad más alta de la nube de puntos (a partir de ahora “Objetivo 2”).

Como por el momento, como se ha analizado, no se consigue unos buenos resultados con la petición de datos continua del LiDAR (instrucción ND), se trabaja con la instrucción de petición por dato (instrucción HD).

OBJETIVO 1:

Para cumplir este objetivo, lo que se hace es dar un margen suficiente de tiempo a la petición de nuevos datos LiDAR, en este caso, pide datos cada 200 milisegundos.

Además, hasta que no se recibe el dato LiDAR, no se “escucha” a los otros dos sensores, y una vez se recibe el dato LiDAR, primero se espera un dato de la unidad de medida inercial y posteriormente un dato del receptor GNSS-RTK. Tras recibir estos dos datos, el programa vuelve a esperar un dato LiDAR.

LiDAR:

En primer lugar se analizan los datos temporales del sensor LiDAR.

En la figura 81 se muestra el tiempo entre recepciones de este sensor.

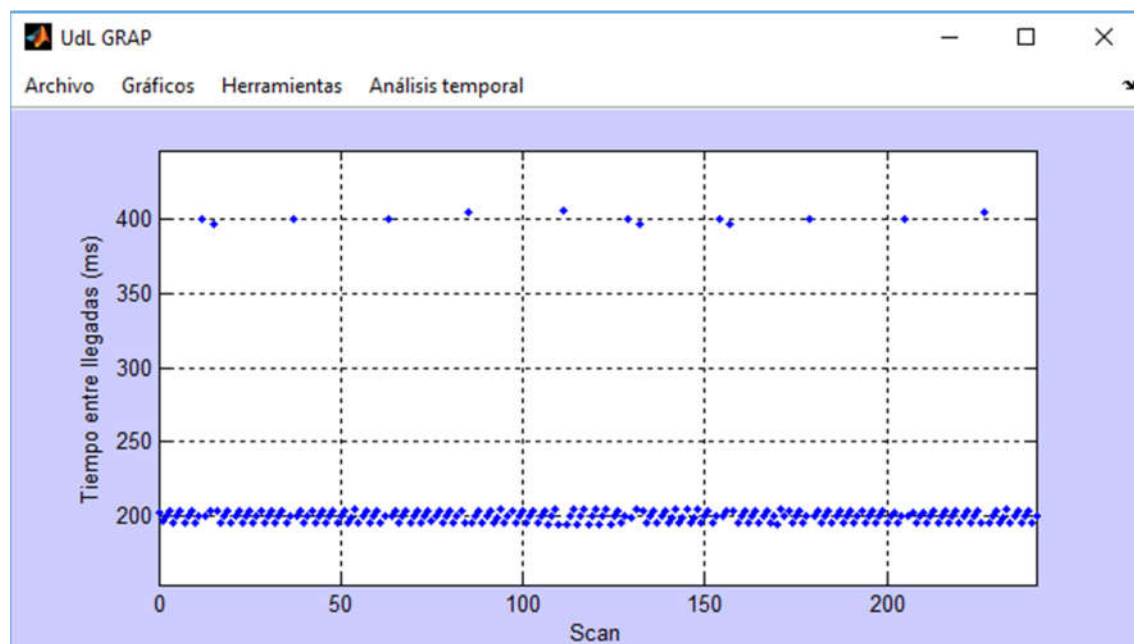


Figura 81: Tiempo entre recepciones del sensor LiDAR [Adq. simult².- Objetivo 1]

² Adquisición simultánea

Como se puede observar, se ha puesto un tiempo de espera a la nueva petición de 200 milisegundos, pero en este caso, esos 200 milisegundos empiezan a contar desde que se envía el comando de petición, por ese motivo el tiempo entre llegadas es tan estable en esa franja.

También se pueden visualizar datos en los 400 milisegundos, estos son a causa de datos no recibidos correctamente, ya sea por error de comunicación, de transferencia de dato, de recepción por parte del microcontrolador o por otras causas.

En la figura 82 se muestra el tiempo de procesado y guardado de los datos LiDAR recibidos.

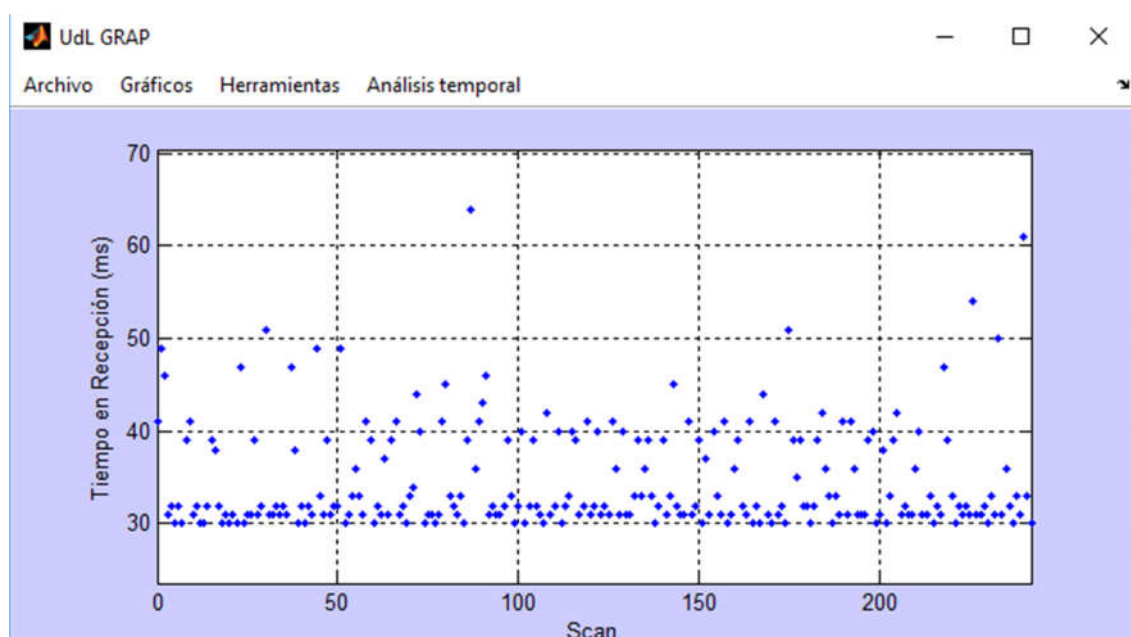


Figura 82: Tiempo de procesado y guardado de datos del sensor LiDAR [Adq. simult.- Objetivo 1]

Como se puede observar, la mayoría de datos se procesan y guardan en un tiempo de entre 30 y 33 milisegundos.

La disparidad de estos datos puede deberse ya sea al tamaño de los datos recibidos, ya que si hay multiretornos o las distancias recibidas son mayores el proceso es más costoso, o a dificultades en la comunicación para el guardado en la tarjeta SD, ya que se entra en ella muchas veces y en las últimas pruebas su funcionamiento no era del todo correcto (ni si quiera probándola en un PC portátil [Asus K555L]).

Por último, en la figura 83 se muestra el “tiempo muerto” del LiDAR, el tiempo entre que acaba de guardar los datos y vuelve a recibir uno nuevo.

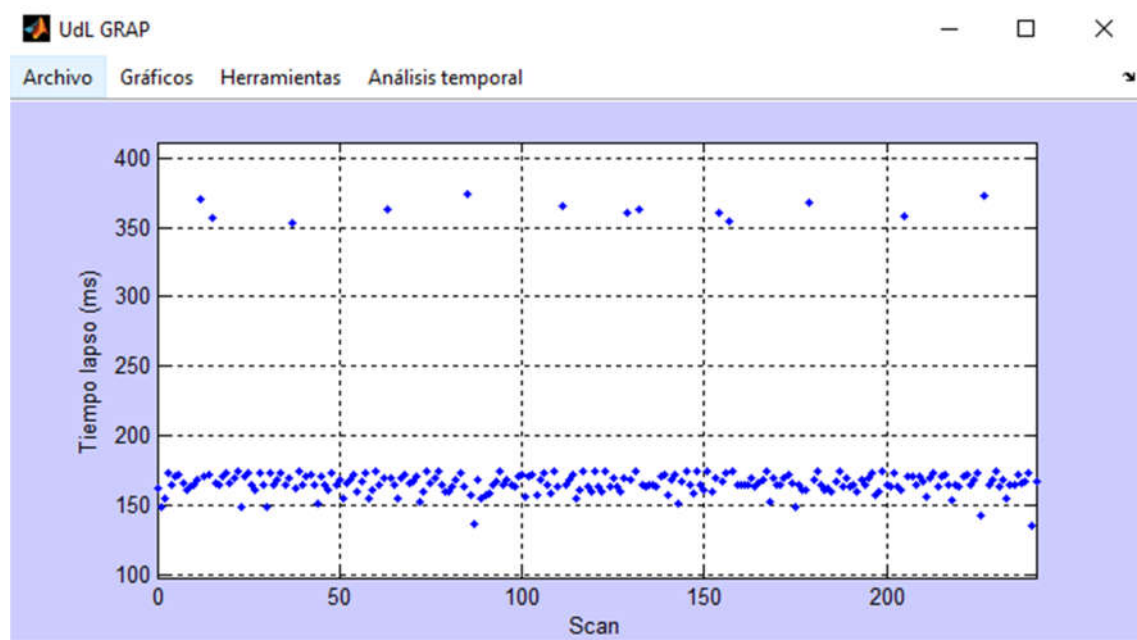


Figura 83: Tiempo muerto del sensor LiDAR [Adq. simult.- Objetivo 1]

Como se puede observar, este tiempo es la resta entre el tiempo entre recepciones y el tiempo de guardado.

Este tiempo muerto, en caso de que la adquisición fuese tan solo de datos LiDAR, sería el que teóricamente se debería de acercar a cero, aunque como se han visto en las pruebas anteriores, si se acerca demasiado a cero llega un momento que la adquisición se bloquea.

De todas formas, con esta configuración se ha conseguido lo que se andaba consiguiendo con todos los ensayos realizados, evitar el bloqueo del microcontrolador, lo cual sirve de punto de partida para nuevas versiones las cuales se centren en mejorar la frecuencia de adquisición.

Unidad de medida inercial:

En segundo lugar, ya que es el sensor del que se adquieren los datos una vez se han procesado y guardado los del sensor LiDAR, se analizan los datos temporales de la unidad de medida inercial.

En la figura 84 se muestra el tiempo entre recepciones de este sensor.

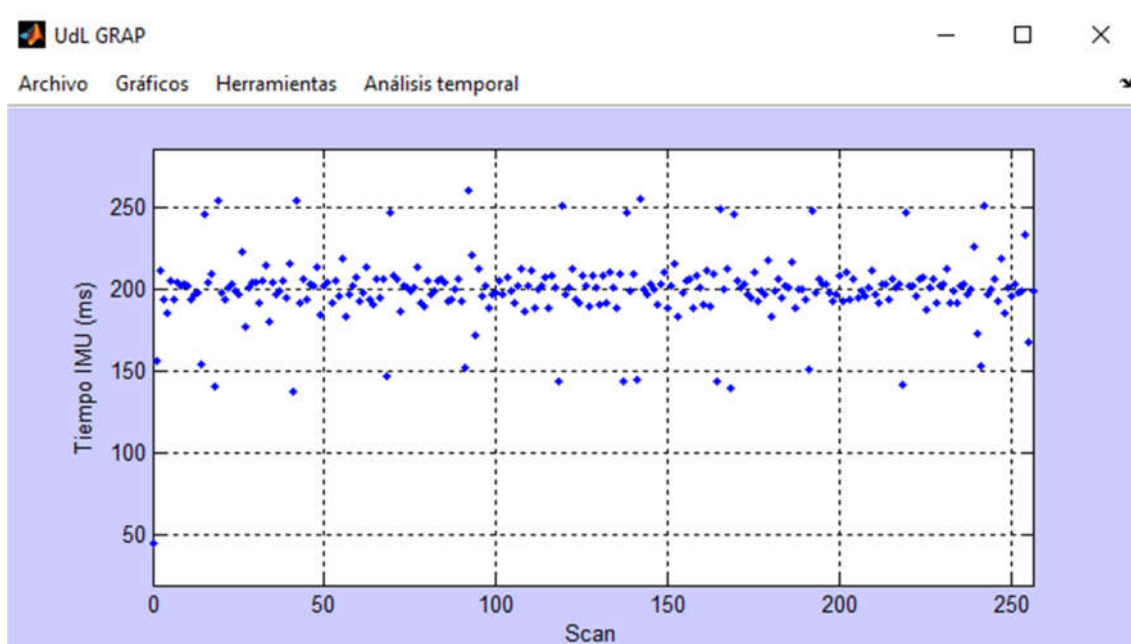


Figura 84: Tiempo entre recepciones de la unidad de medida inercial [Adq. simult.- Objetivo 1]

En la figura 85 se amplían la zona de interés de los datos de la figura 84 para facilitar su visualización y análisis.

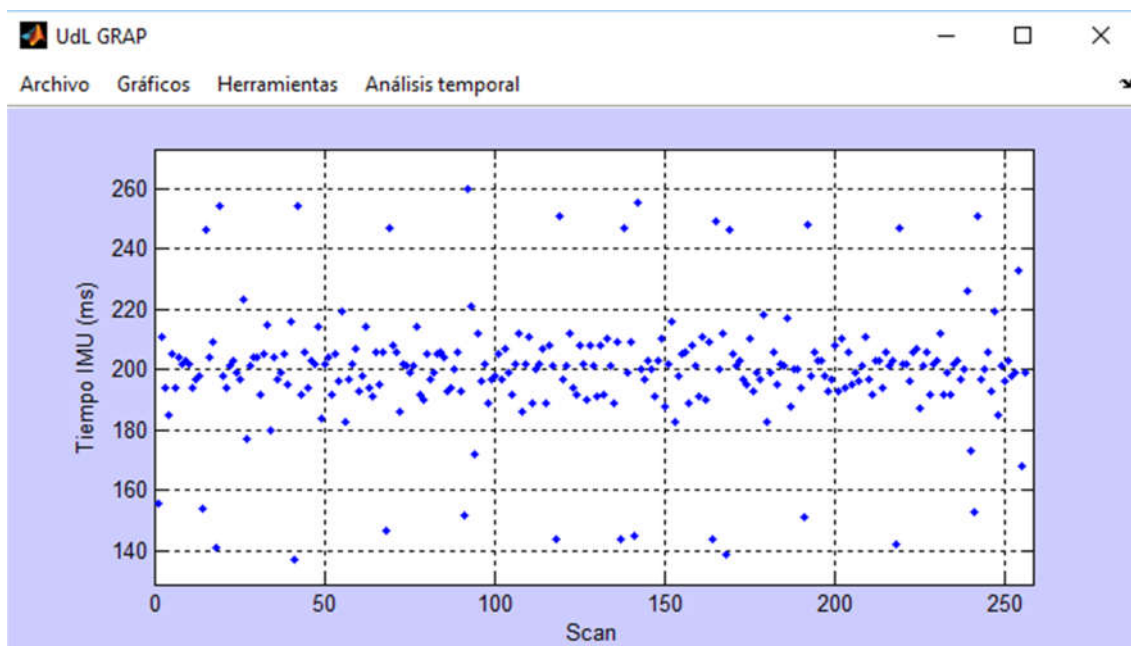


Figura 85: Ampliación del eje Y de la gráfica de la figura 80

Se puede observar que los datos de la unidad de medida inercial se concentran en la franja de 200 milisegundos.

Esto es normal ya que si el tiempo de procesado y guardado de los datos LiDAR es estable, el tiempo en el que se comenzará a “escuchar” a la unidad de medida inercial será estable.

Los saltos en la gráfica de tiempos que se está analizando son debidos a inestabilidades del tiempo de guardado de los datos del sensor LiDAR sumadas a los posibles tiempos de espera que pueda causar la recepción de los datos de la unidad de medida inercial, aunque éstos, en teoría, son mínimos o nulos, ya que solo existirían si en el momento de “escuchar” al sensor este justamente está a mitad de envío de un dato nuevo, y este tiempo es despreciable en el análisis.

GNSS-RTK:

En último lugar se analiza el receptor GNSS-RTK, del cual se espera el dato tras recibir tanto el dato LiDAR como posteriormente el dato de la unidad de medida inercial.

En la figura 86 se muestra el tiempo entre recepciones de este sensor.

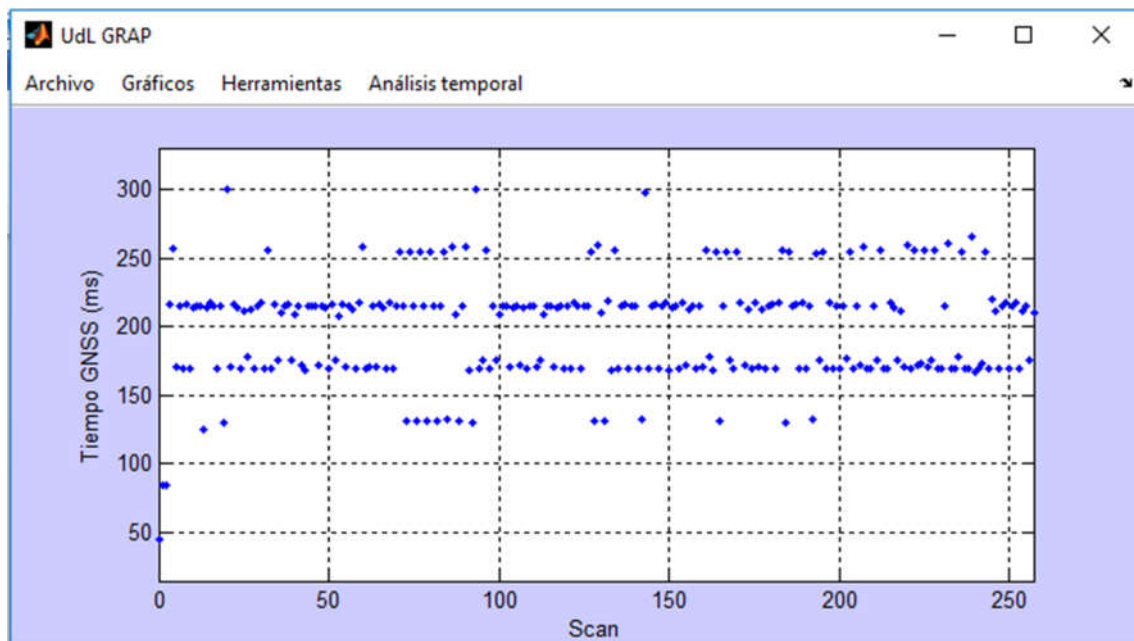


Figura 86: Tiempo entre recepciones del receptor GNSS-RTK [Adq. simult.- Objetivo 1]

En la figura 87, se muestra una ampliación de la zona más significativa de la figura 86, para facilitar la visualización y análisis de la gráfica obtenida.

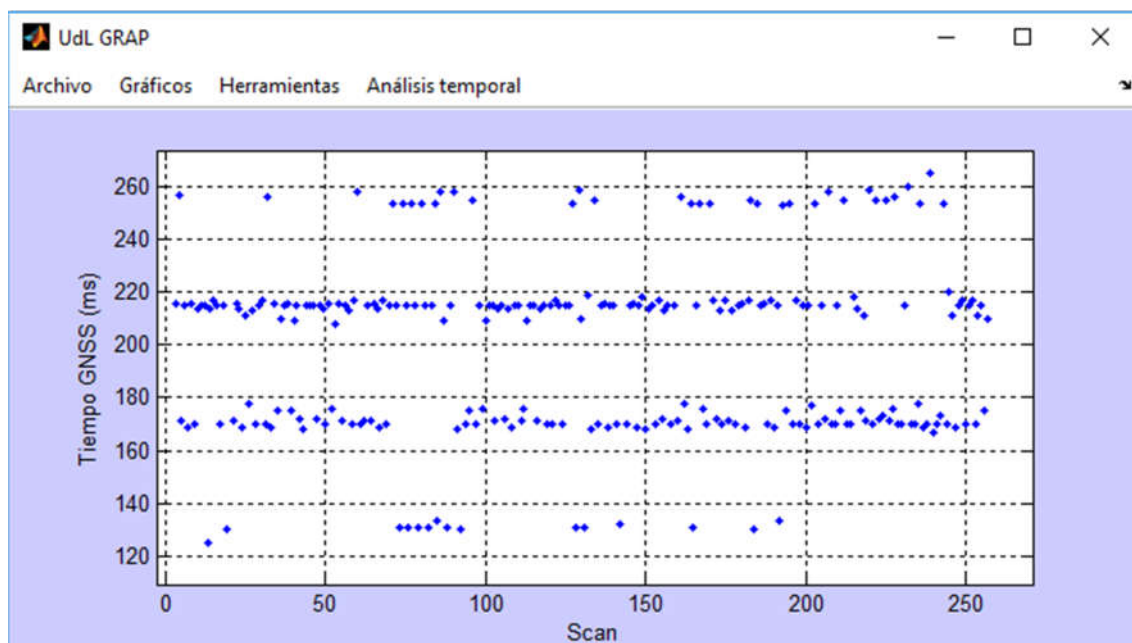


Figura 87: Ampliación del eje Y de la gráfica de la figura 86.

Se puede observar una gráfica algo dispar con 4 zonas de tiempo bien definidas.

Las zonas superior e inferior parecen debidas a pérdidas de datos del propio sensor, ya que si en una lectura pierde un dato, pierde 40-50 milisegundos, pero en la siguiente lectura, si no pierde dato, esos 40-50 milisegundos son de adelanto.

Las dos zonas centrales pueden ser debidas a un efecto parecido al anterior, pero en vez de debido a la pérdida de datos, puede ser debido a la suma de defectos temporales en el proceso y guardado de los sensores anteriores, sumado a la espera de recepción de dato de éste.

En todo caso, son datos que no alteran la estabilidad del sistema ni hacen que este se bloquee.

El problema que pueden conllevar estos datos temporales es la falta de precisión de los datos recogidos, ya que si se desvían demasiado, no está posicionando el dispositivo de adquisición en el momento de la lectura del sensor LiDAR, sino un instante después.



Si la velocidad de desplazamiento es baja, ese instante puede ser más amplio y no significar tanto error, pero si la velocidad de desplazamiento es alta, ese instante puede significar una reconstrucción de la nube inservible.

OBJETIVO 2

Para este objetivo se utiliza una versión anterior del programa, en la cual los datos son adquiridos aleatoriamente, es decir, cada vez que llega el dato de un sensor se lee y guarda ese dato, sin dar prioridad a ninguno de los tres sensores.

Además, la petición de dato para el sensor LiDAR es cada 35 milisegundos.

El programa se bloquea en un tiempo aleatorio, pero nunca mayor de un minuto.

LiDAR:

En primer lugar se hará un análisis temporal de la adquisición de datos del sensor LiDAR.

La figura 88 muestra el tiempo entre recepciones de este sensor.

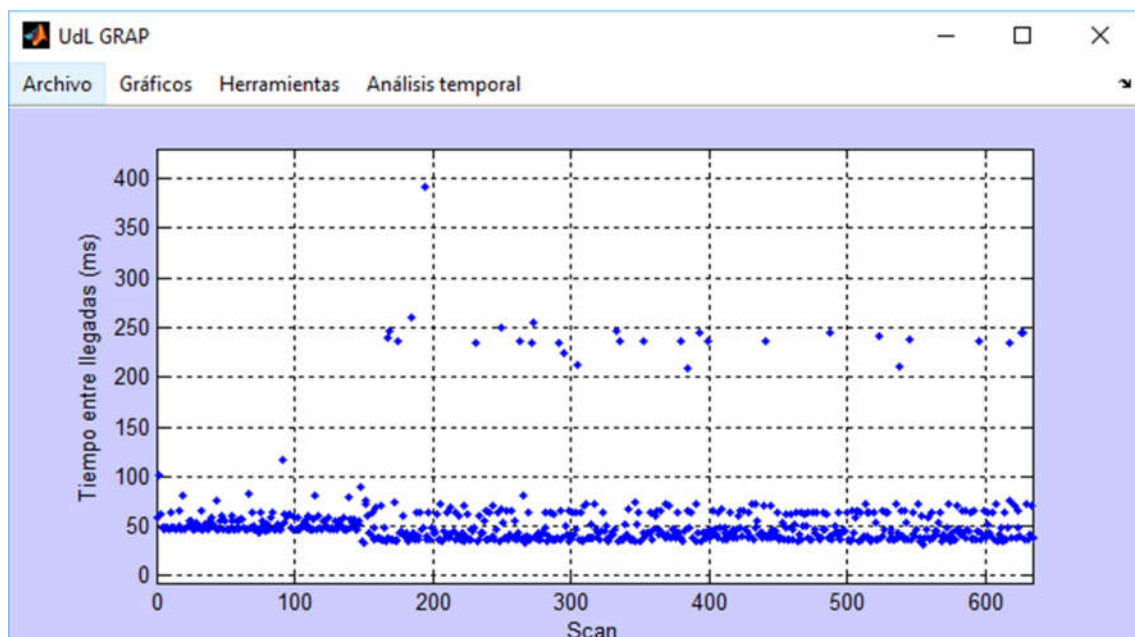


Figura 88: Tiempo entre recepciones del sensor LiDAR [Adq. simult.- Objetivo 2]

Se puede observar que las llegadas se concentran en los 50 milisegundos, habiendo algunas muestras que se colocan entre los 200 y los 250 milisegundos.

En la figura 89 se amplía la zona de mayor población de muestras, la zona de entre los 30 y 80 milisegundos, para poder analizar con mayor claridad lo que allí ocurre.

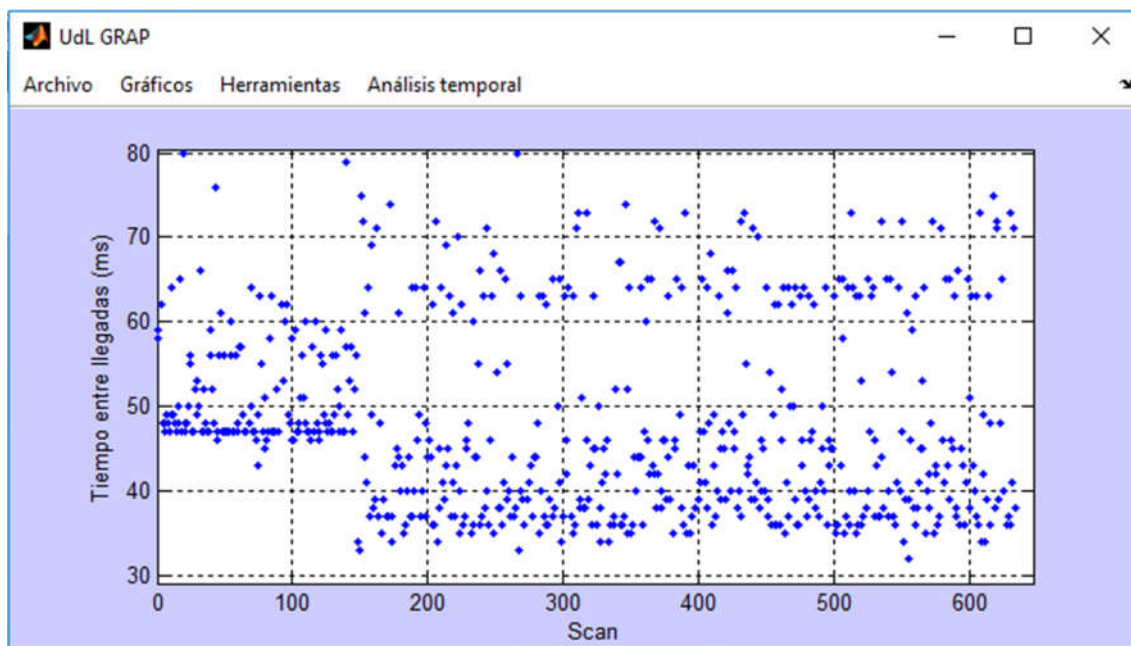


Figura 89: Ampliación del eje Y de la gráfica de la figura 88

De estas muestras temporales se puede concluir que tenemos una parte de las muestras que tienden hacia los 35 milisegundos y otra que tiene hacia los 70. Esto parece que se debe a pérdidas de algunos datos, y recepción del siguiente, ya que, como se ha dicho, la comanda HD para recepción de dato se realiza cada 35 milisegundos.

De todas formas, son unas muestras muy dispersas que no muestran un funcionamiento estable del programa de adquisición.

En la figura 90, se muestra el tiempo de procesado y guardado de las muestras del sensor LiDAR.

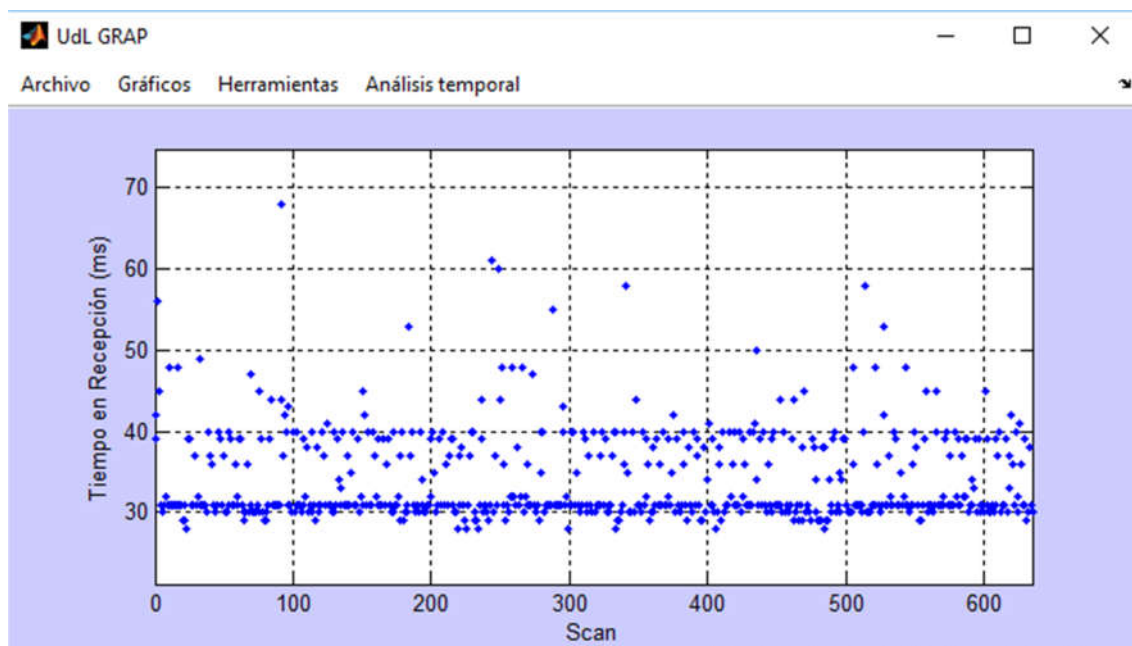


Figura 90: Tiempo de procesado y guardado de datos del sensor LiDAR [Adq. simult – Objetivo 2]

Como se puede apreciar, el tiempo tiende a ser de 30 milisegundos, no llegando a sobrepasar la mayoría de las muestras los 40 milisegundos.

Este tiempo parece ser estable en todos los análisis realizados hasta el momento.

En la figura 91 se muestra el “tiempo muerto” del sensor LiDAR.

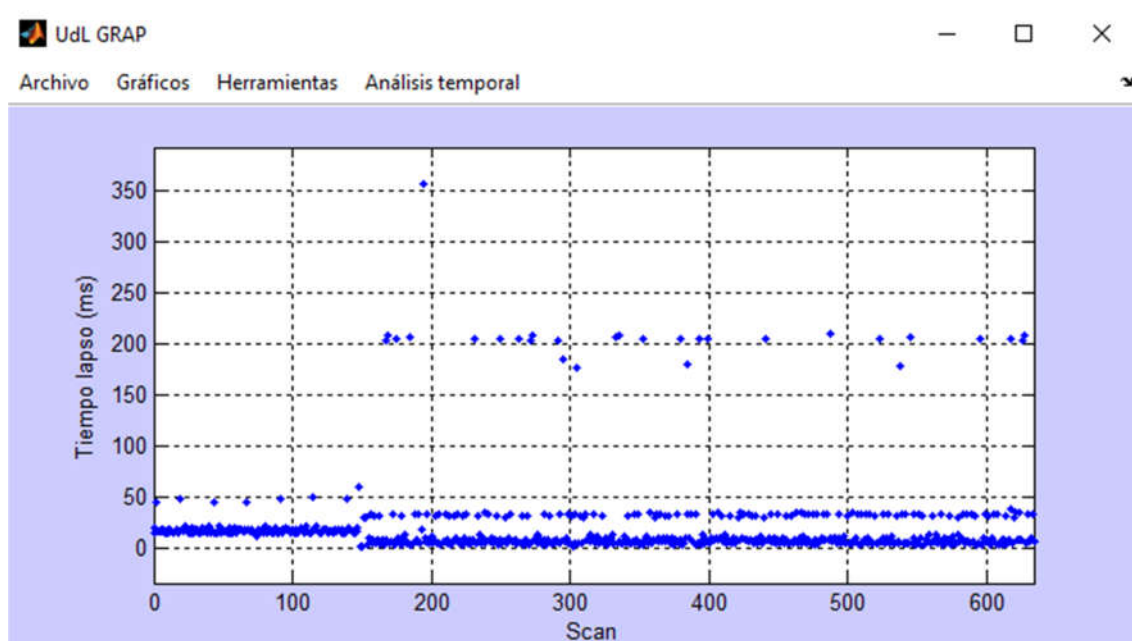


Figura 91: Tiempo muerto del sensor LiDAR [Adq. simult.- Objetivo 2]

Como se puede observar, hay momentos en los que este tiempo está sobre el cero. Esto se puede considerar una causa para el bloqueo del código al saturar la memoria de órdenes sin poder gestionar tantas órdenes a la vez.

Unidad de medida inercial:

En segundo lugar se analiza el comportamiento temporal de las muestras de la unidad de medida inercial.

En la figura 92 se muestra el tiempo entre recepciones de este sensor.

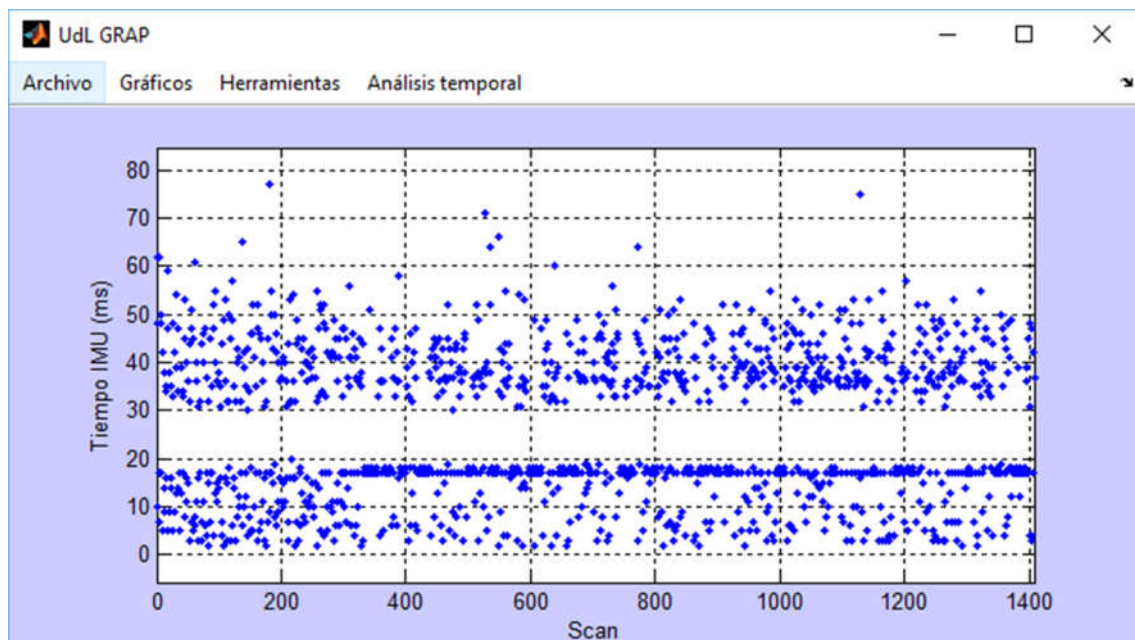


Figura 92: Tiempo entre recepciones de la unidad de medida inercial [Adq. simult. - Objetivo 2]

Como ha sido expuesto anteriormente en el análisis individual del sensor, el tiempo entre recepciones es muy estable entre los 17 y los 18 milisegundos.

Como se puede observar en estos datos, los tiempos de adquisición de datos del sensor son muy inestables, lo cual no es una buena señal a la hora de garantizar la estabilidad del sistema de adquisición.

Receptor GNSS-RTK:

Por último se analiza el comportamiento temporal de las muestras obtenidas del receptor GNSS-RTK.

En la figura 93 se muestra el tiempo entre recepciones de este sensor.

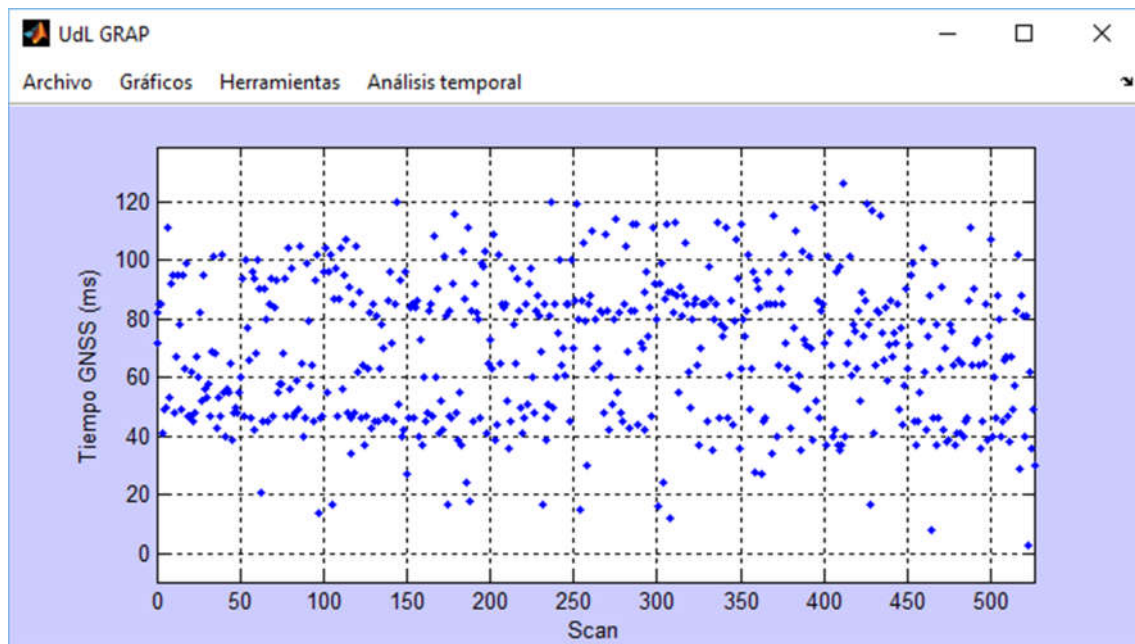


Figura 93: Tiempo entre recepciones del receptor GNSS-RTK [Adq. simult.- Objetivo 2]

Como ha sido expuesto anteriormente en el análisis individual del sensor, el tiempo entre recepciones debería estar entre los 40 y 50 milisegundos (80 milisegundos en caso de pérdida de datos).

Como se puede observar en estos datos, los tiempos de adquisición de datos del sensor son muy inestables, al igual que los de la unidad de medida inercial, lo cual tampoco es una buena señal a la hora de garantizar la estabilidad del sistema de adquisición.

3.2. Nubes de puntos

Respecto a la reconstrucción de las nubes de puntos, se mostrarán algunos resultados para realizar la comparación entre el objetivo 1 y el objetivo 2 descritos en el apartado anterior (3.1. Tiempos de adquisición).

Evidentemente, ambos objetivos tienen como objetivo común el realizar una reconstrucción de la nube de puntos correcta y precisa.

Objetivo 1:

A modo de recordatorio, este objetivo trata de evitar el bloqueo del sistema de adquisición manteniéndolo estable durante un periodo de tiempo indefinido.

Mantenerlo estable durante un tiempo indefinido sirve, por ejemplo, para obtener resultados como el de la figura 94, donde sin problema se ha dado la vuelta poco a poco al coche de validación.

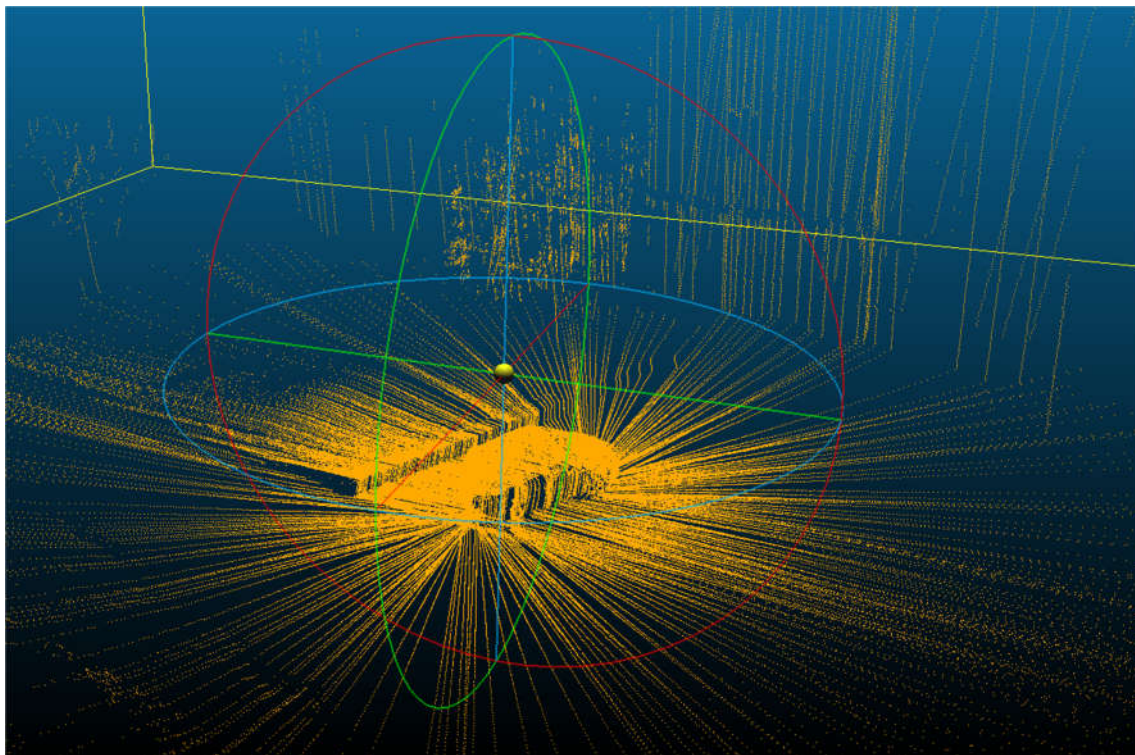


Figura 94: Nube de puntos, rotación completa al vehículo [Objetivo 1]

O para dar poco a poco un giro de 360° para obtener un resultado como el de la figura 95.

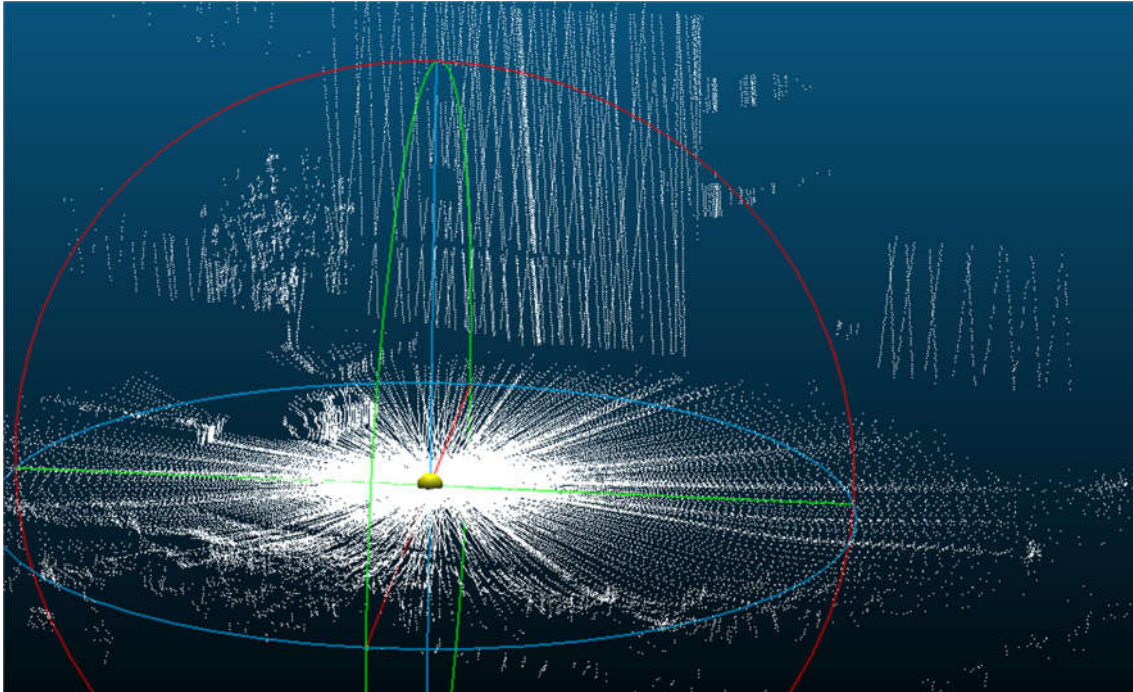


Figura 95: Nube de punto, giro yaw de 360° desde punto fijo [Objetivo 1]

Y para comprobar que el sensor inercial no solo trabaja bien en su giro yaw, se realiza un escaneo del lateral del coche girando el ángulo pitch, como se muestra en la figura 96.

Aunque no se aprecie muy bien, es una muy buena reconstrucción del mismo.

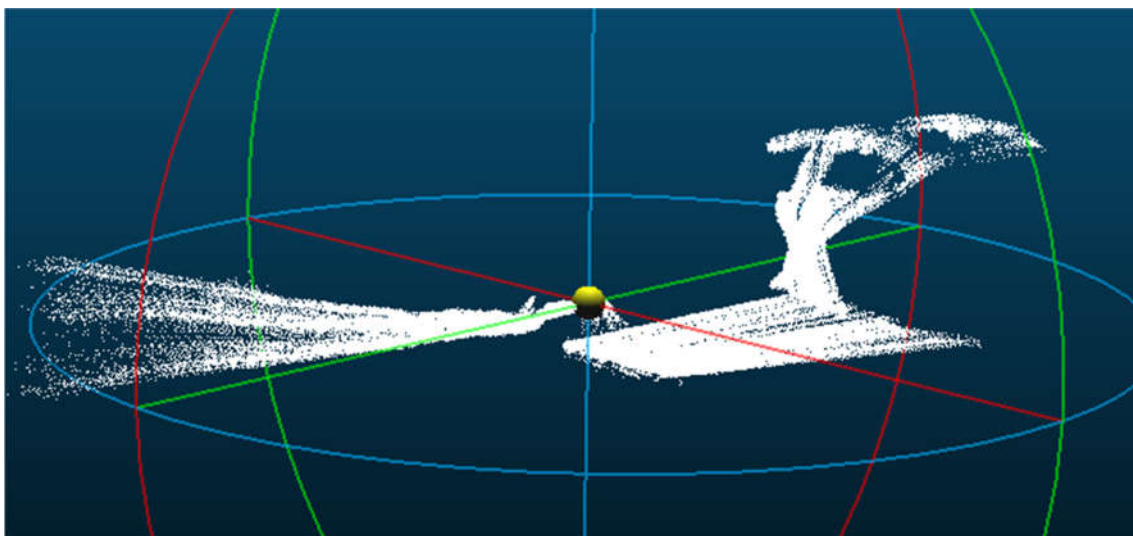


Figura 96: Nube de puntos, giro pitch desde punto fijo [Objetivo 1]

Y por último en la figura 97 se puede apreciar el lateral izquierdo del coche y el escalón del jardín perfectamente definido. Y el lateral derecho en la figura 98.

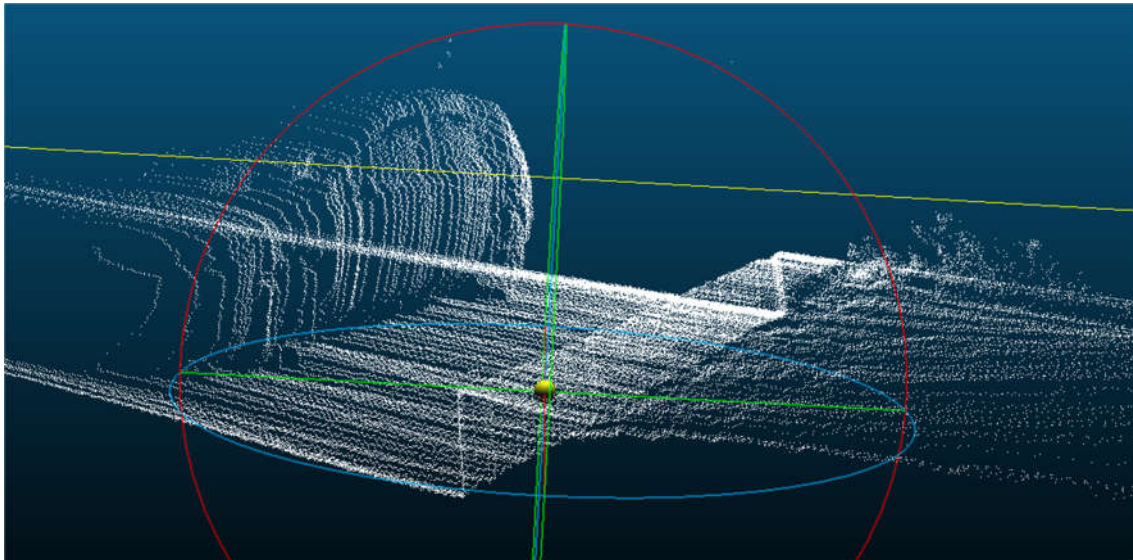


Figura 97: Nube de puntos, lateral izquierdo vehículo [Objetivo 1]

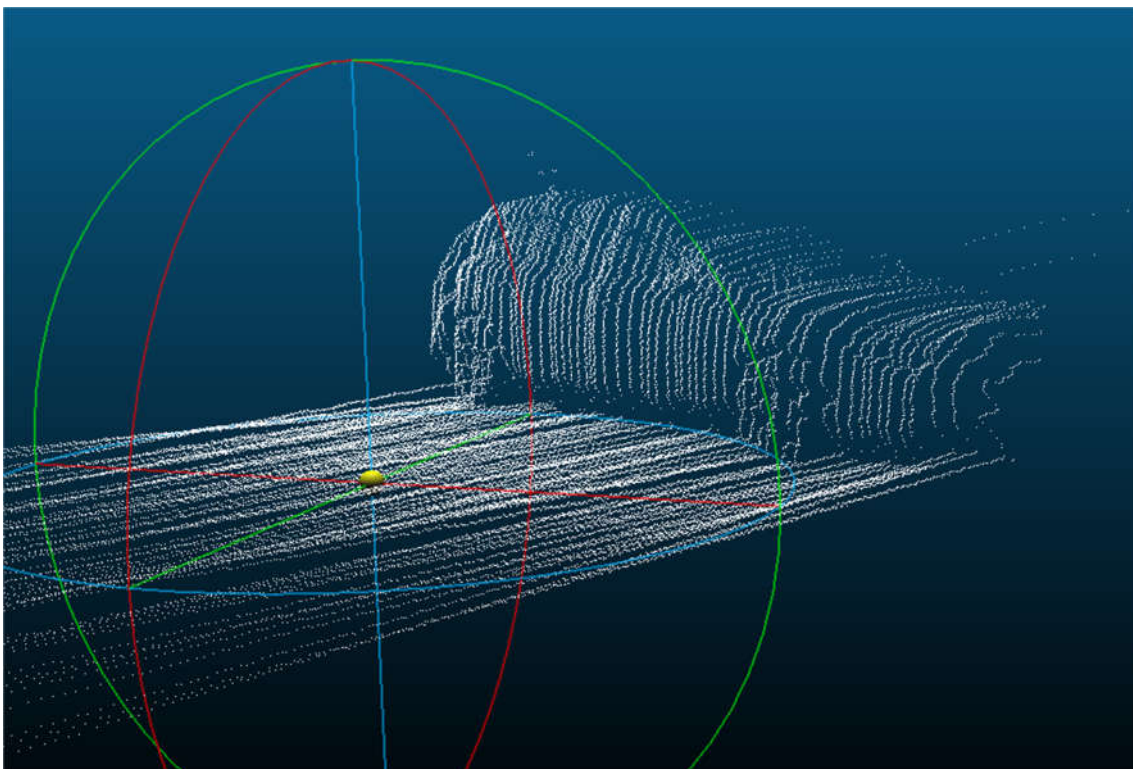


Figura 98: Nube de puntos, lateral derecho vehículo [Objetivo 1]

Una vez comprobado que objetivo 1 cumple el requisito de no bloquearse, y habiendo validado cualitativamente la reconstrucción de la nube de puntos de forma positiva, se debe aplicar la *prueba de validación 2*, descrita en el apartado 2.3.4 *Validación de la nube de puntos*.

Los resultados de las pruebas realizadas con el software CloudCompare, con el objetivo de validar dimensionalmente la reconstrucción realizada, teniendo en cuenta que las medidas han sido realizadas de forma manual por el usuario, son:

- Longitud: 4,268 m (real 4,25 m)
- Altura: 1,442 m (real 1,46 m)
- Anchura: 1,698 m (real 1,75 m)
- Batalla: 2,595 m (real 2,61 m)

Los resultados ofrecidos por la prueba indican que, dimensionalmente, la reconstrucción es afín a la realidad.

En definitiva, las reconstrucciones de las nubes de puntos son correctas, el microcontrolador no se bloquea durante la adquisición, pero la frecuencia de adquisición es demasiado baja.

Objetivo 2:

Este objetivo es el de recoger el máximo número de datos para la reconstrucción de una nube lo más densa posible.

En la figura 99 se puede apreciar el lateral derecho del coche de validación y en la figura 100 se puede apreciar su lado derecho.

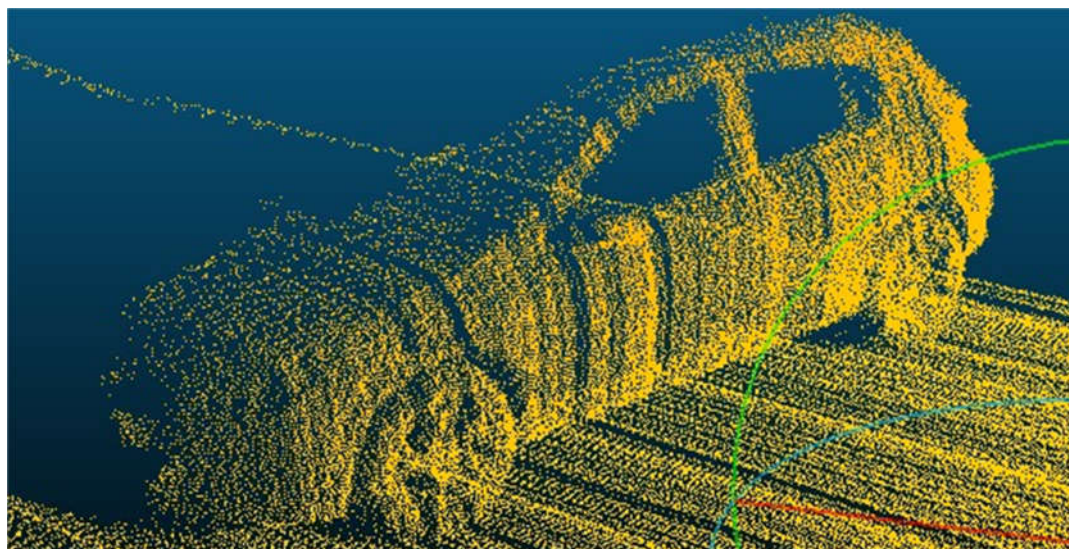


Figura 99: Nube de puntos, lateral izquierdo delantero vehículo [Objetivo 2]

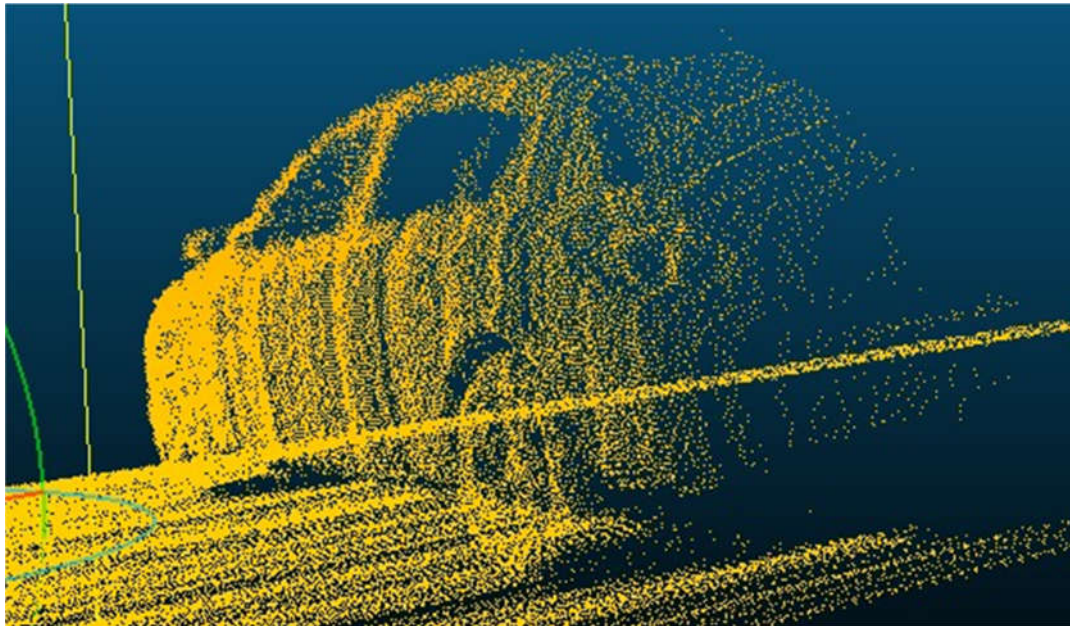


Figura 100: Nube de puntos, lateral izquierdo trasero vehículo [Objetivo 2]

Se puede ver como la reconstrucción es densa y cualitativamente precisa, el único problema es que tan solo da tiempo a realizar un escaneo de estas dimensiones, ya que no da tiempo a dar una vuelta entera ni si quiera al coche de validación.

Los resultados de las pruebas realizadas con el software CloudCompare, con el objetivo de validar dimensionalmente la reconstrucción realizada, teniendo en cuenta que las medidas han sido realizadas de forma manual por el usuario, son:

- Longitud: 4,248 m (real 4,25 m)
- Altura: 1,432 m (real 1,46 m)
- Anchura: No comprobable³.
- Batalla: 2,581 m(real 2,61 m)

Los resultados ofrecidos por la prueba indican que, dimensionalmente, la reconstrucción es afín a la realidad.

En definitiva, las reconstrucciones de las nubes de puntos son correctas, la frecuencia de adquisición es la adecuada, pero el microcontrolador se bloquea durante la adquisición.

³ En el ensayo se tuvo problemas con las medidas de la unidad de medida inercial al tomar las medidas con la estructura inicial. Éstos han sido descritos en el apartado 2.2.1. *Adquisición*

En la figura 101 se aprecian las dos figuras geométricas de validación escaneadas tan solo desde uno de sus lados.

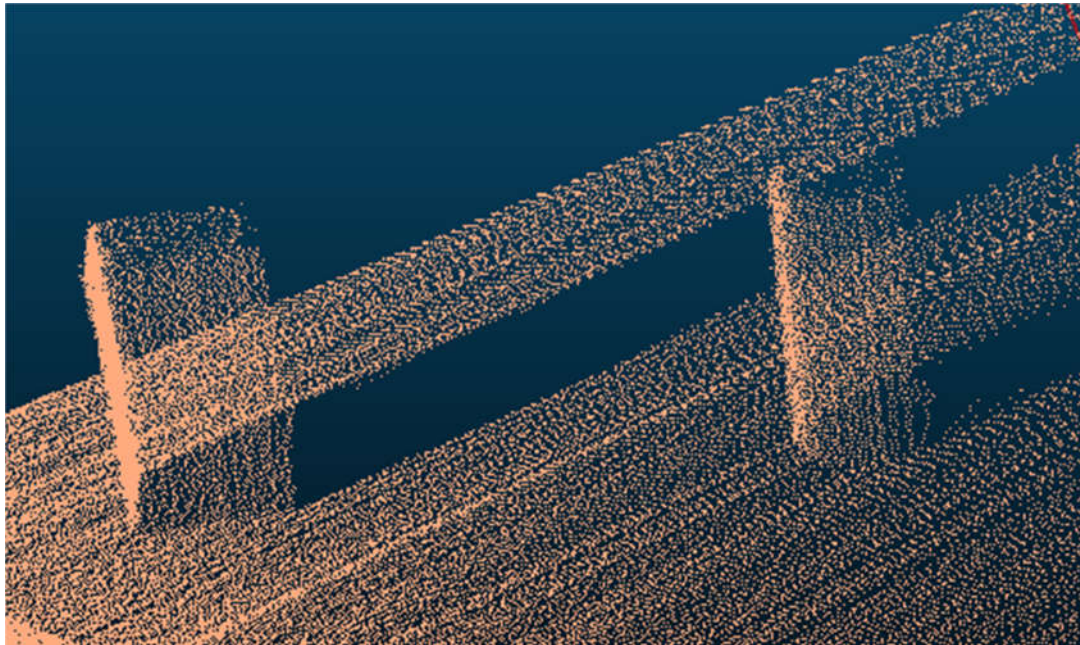


Figura 101: Nube de puntos, figuras geométricas [Objetivo 2]

Los resultados de las pruebas realizadas con el software CloudCompare, con el objetivo de validar dimensionalmente la reconstrucción realizada, teniendo en cuenta que las medidas han sido realizadas de forma manual por el usuario, son:

Cilindro:

Altura: 0.971 m (real 0.95 m)

Diámetro: 0.478 (real 0.49 m)

Prisma rectangular:

Altura: 0.973 m (real 1.00 m)

Lado 1: 0.527 (real 0.50 m)

Lado2: 0.538 (real 0.50 m)

Los resultados ofrecidos por la prueba indican que, dimensionalmente, la reconstrucción es afín a la realidad.

Con el objetivo 2 se ven unas reconstrucciones de nubes de puntos 3D con una relación densidad/velocidad de desplazamiento aceptable, pero el dispositivo es inútil si tan solo puede escanear durante unos segundos.

Por otro lado, con el objetivo 1 se puede escanear durante, teóricamente, todo el tiempo que se quiera hasta que se agote la memoria SD, pero para conseguir una densidad de puntos



semejantes a la conseguida con el objetivo 2 o mayor, se necesitaría realizar la operación de escaneo con un desplazamiento significativamente más lento.

En ambos casos, la reconstrucción es buena, aunque la reconstrucción del objetivo 1 sería mejor si se cogieran una cantidad mayor de datos por unidad de tiempo.



4. Conclusiones

- Se ha conseguido que los tres sensores trabajen sincronizados en la adquisición de datos sin bloqueos.
- Se ha mejorado el tiempo de procesado y guardado de cada uno de los sensores.
- Se ha conseguido un script de Matlab que reconstruya la nube de puntos a partir de los datos guardados en la memoria SD para poder visualizarlos con CloudCompare.
- Se ha conseguido una interface gráfica de Matlab con la cual visualizar los datos obtenidos y analizarlos de forma cómoda.
- Se ha realizado un código de color con los led de la placa para conocer el estado del programa de adquisición en cualquier momento.
- Se debe ahondar en el error del comando de petición continua de datos "ND" del LiDAR, ya que debería reducir errores de comunicación al no tener que realizar peticiones constantemente.
- Se debe de reducir el tiempo entre peticiones del comando HD del LiDAR utilizado en el objetivo 1 para intentar aproximarse a la densidad de puntos conseguida por el objetivo 2.

En general, se han conseguido los objetivos del proyecto, la reconstrucción es de buena calidad y el bloqueo del sistema ha desaparecido. Tras muchos análisis y errores de cada uno de los dispositivos utilizados, así como en los algoritmos de reconstrucción, se ha conseguido un sistema estable y un programa de reconstrucción cómodo y sencillo de utilizar.

No obstante, es cierto que queda un último tramo que recorrer para conseguir una mayor precisión y una mayor densidad en la nube de puntos, pero con los análisis realizados, al menos, se ha allanado el camino hacia los siguientes objetivos.



5. Propuestas de mejora

- Guardar los datos recibidos del LiDAR sin procesamiento previo, para lo cual hay que conseguir realizar un código (ya iniciado) que lea la tarjeta SD, busque una a una cada memoria guardada, las lea, las procese y vuelva a guardar los datos con otro nombre en la misma memoria pero con el formato estipulado para el funcionamiento actual del programa de posprocesamiento.
- Reducir al máximo el tiempo de petición del comando HD.
- Realizar una estructura + mochila cómodas para el escaneo manual durante un largo período de tiempo.
- Trabajar con una ampliación de memoria RAM externa con el objetivo de evitar los bloqueos cuando se trabaja con demasiados datos. Con esto, por ejemplo, se podrían realizar guardados de datos en una sola escritura y así evitar entrar tantas veces a la memoria, como se pretendía con la prueba de la concatenación de datos.



6. Bibliografia

- [1] Escolà, A.; Martínez-Casasnovas, J.A.; Rufat, J.; Arnó, J.; Arbonés, A.; Sebé, F.; Pascual, M.; Gregorio, E.; Rosell-Polo, J.R. 2017. Mobile terrestrial laser scanner applications in precision fruticulture/horticulture and tools to extract information from canopy point clouds. *Precision Agriculture* 18(1), 111-132.
- [2] Solans A., “Sistema de adquisición de datos portátil para la generación de nubes de puntos 3D georreferenciadas a partir de un sensor LiDAR 2D (Parte I, Estudio y Comunicación)”, Trabajo final de carrera, Grup de Recerca en AgròTICa i Agricultura de Precisió, Universitat de Lleida, 2015.
- [3] UTM-30LX-EW Communication Protocol Specification, C-42-03886-E. Hokuyo Automatic Co.,Ltd., Osaka-Japan, 2009.
- [4] Scanning Laser Range Finder UTM-30LX-EW Specification, C-42-3785. Hokuyo Automatic Co.,Ltd., Osaka-Japan, 2012.
- [5] MT Low-Level Communication Protocol Documentation, MT0101P. Xsens Technologies B.V., Los Angeles, 2010.
- [6] MTi and MTx User Manual and Technical Documentation, MT0100P. Xsens Technologies B.V., Los Angeles-California, 2010.
- [7] Morales D., “Puesta en marcha del Sensor Inercial MTi de Xsens”, Trabajo final de carrera, Dep. Ingeniería de sistemas y automática, Universidad Carlos III de Madrid, Madrid, 2011.
- [8] Leica GPS1200+ Technical Reference, V 8.5. Leica Geosystems AG, Heerburgg-Switzerland, 2011.
- [9] STM32F405xx-STM32F407xx, 022152 Rev 4. STMicroelectronics, Geneva - Switzerland, 2013
- [10] RM0090 Reference manual, 018909 Rev 4. STMicroelectronics ,Geneva - Switzerland, 2013
- [11] STM32F4DIS-BB User Manual, REV 1.0. Embest Technology Co.,LTD, New Taipei City – Taiwan, 2012.



7. Anejos

7.1. Códigos de adquisición, preprocesamiento y guardado de datos

Son los códigos utilizados por la placa STM32F4-discovery, del fabricante STM, para la adquisición, preprocesamiento y guardado de datos de los distintos sensores.

A continuación se presentan las últimas versiones de cada código, las cuales se encuentran adjuntas en la carpeta de archivos de este documento. El funcionamiento de estas versiones de los códigos ha sido explicado en el apartado 2.2. *Adquisición, preprocesamiento y almacenamiento de señales simultáneas*.

- Las versiones con terminación “a” realizan la adquisición, el preprocesamiento de datos y el guardado de éstos en la tarjeta SD.
- Las versiones con terminación “b” realizan la adquisición y guardan los datos en bruto, sin un procesado previo. En caso de las versiones “FULL”, no preprocesan los datos LiDAR pero sí los procedentes de la IMU y el receptor GNSS-RTK.
- Las versiones con terminación “c” (solo presentes en las versiones “FULL”), no preprocesan los datos ni de LiDAR ni de IMU.
- Las acciones realizadas con los datos recibidos del receptor GNSS-RTK no son consideradas como preprocesamiento, sino como un simple filtrado de datos, con lo que por un lado no tienen subíndice “a, b” y por otro, en caso de adquisición de datos simultánea con otros sensores, no modifican su comportamiento ante sus subíndices.

Códigos FULL:

Son los códigos en los que se realiza la adquisición simultánea de los tres sensores.

- STM32F4-FULL_HD v3.2a
Utiliza el comando HD para la petición de lecturas individuales
- STM32F4-FULL_HD v3.2b
Utiliza el comando HD para la petición de lecturas individuales



- STM32F4-FULL_HD v3.2c

Utiliza el comando HD para la petición de lecturas individuales

- STM32F4-FULL_ND v3.2a

Utiliza el comando ND para la posterior recepción de datos continuos.

- STM32F4-FULL_ND v3.2b

Utiliza el comando ND para la posterior recepción de datos continuos.

- STM32F4-FULL_ND v3.2c

Utiliza el comando ND para la posterior recepción de datos continuos.

Códigos GPS:

Son los códigos en los que se realiza la adquisición individual del receptor GNSS-RTK.

- STM32F4-GPS v3.2

Códigos IMU:

Son los códigos en los que se realiza la adquisición individual de la unidad de medida inercial.

- STM32F4-IMU v3.2a
- STM32F4-IMU v3.2b

Códigos IMUGPS:

Son los códigos en los que se realiza la adquisición simultánea del receptor GNSS-RTK y la unidad de medida inercial.

- STM32F4-IMUGPS v3.2a
- STM32F4-IMUGPS v3.2b



Códigos LiDAR:

Son los códigos en los que se realiza la adquisición individual del sensor LiDAR.

- STM32F4-LIDAR_HD v3.2a

Utiliza el comando HD para la petición de lecturas individuales.

- STM32F4-LIDAR_HD v3.2b

Utiliza el comando HD para la petición de lecturas individuales.

- STM32F4-LIDAR_ND v3.2^a

Utiliza el comando ND para la posterior recepción de datos continuos.

- STM32F4-LIDAR_ND v3.2a - sin guardar datos

Utiliza el comando ND para la posterior recepción de datos continuos, pero no realiza ni procesado ni guardado de los mismos, tan solo guarda los tiempos entre llegada y llegada de éstos.

- STM32F4-LIDAR_ND v3.2b

Utiliza el comando ND para la posterior recepción de datos continuos.



7.2. Códigos para el posprocesamiento de datos

Son los códigos que se pueden utilizar en cualquier PC con los programas R-Studio y Matlab instalados para realizar un análisis de los datos obtenidos o bien para llevar a cabo la reconstrucción de la nube de puntos a partir de éstos.

Códigos R-Studio

- R processing_v.2

Realiza la adaptación de la matriz de datos guardada en la tarjeta SD para su posterior utilización con Matlab. Su función es convertir una matriz “irregular” en una matriz cuadrada o rectangular (mismas dimensiones de todos los vectores que la forman).

Códigos Matlab

- LIDAR_v3_0

Código utilizado para el análisis de los datos recogidos individualmente del sensor LiDAR.

- IMU_v3_0

Código utilizado para el análisis de los datos recogidos individualmente de la unidad de medida inercial.

- GNSS_v3_0

Código utilizado para el análisis de los datos recogidos individualmente del receptor GNSS-RTK.

- Nube_3D_v3_1

Código utilizado para el análisis de los datos recogidos de los tres sensores simultáneos, aunque también puede servir para el análisis de datos recogidos por los sensores individualmente o por dos sensores simultáneos.

También realiza la reconstrucción de la nube de puntos, así como el guardado de ésta para su posterior utilización.